High-dimensional Sparse Count Data Clustering Using Finite Mixture Models

Nuha Zamzami

A Thesis

in

The Concordia Instititute

for

Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Information and Systems Engineering) at

Concordia University

Montréal, Québec, Canada

November 2019

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the thesis prepared

By: Nuha Zamzami

Entitled: High-dimensional Sparse Count Data Clustering Using Finite Mixture Models

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy (Information Systems Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Akshay Kumar Rathore

_____ External Examiner
Dr. Moamar Sayed-Moucheweh

_____ External to Program
Dr. Lyes Kadem

_____ Examiner
Dr. Abdessamad Ben Hamza

_____ Examiner
Dr. Amr Youssef

_____ Thesis Supervisor
Dr. Nizar Bouguila

Approved by _____
Dr. Mohammad Mannan, Graduate Program Director

January 22, 2020 _____
Dr. Amir Asif, Dean
Gina Cody School of Engineering & Computer Science

Abstract

High-dimensional Sparse Count Data Clustering Using Finite Mixture Models

Nuha Zamzami, Ph.D.

Concordia University, 2019

Due to the massive amount of available digital data, automating its analysis and modeling for different purposes and applications has become an urgent need. One of the most challenging tasks in machine learning is clustering, which is defined as the process of assigning observations sharing similar characteristics to subgroups. Such a task is significant, especially in implementing complex algorithms to deal with high-dimensional data. Thus, the advancement of computational power in statistical-based approaches is increasingly becoming an interesting and attractive research domain. Among the successful methods, mixture models have been widely acknowledged and successfully applied in numerous fields as they have been providing a convenient yet flexible formal setting for unsupervised and semi-supervised learning. An essential problem with these approaches is to develop a probabilistic model that represents the data well by taking into account its nature. Count data are widely used in machine learning and computer vision applications where an object, *e.g.,* a text document or an image, can be represented by a vector corresponding to the appearance frequencies of words or visual words, respectively. Thus, they usually suffer from the well-known curse of dimensionality as objects are represented with high-dimensional and sparse vectors, *i.e.,* a few thousand dimensions with a sparsity of 95 to 99%, which decline the performance of clustering algorithms dramatically. Moreover, count data systematically exhibit the burstiness and overdispersion phenomena, which both cannot be handled with a generic multinomial distribution, typically used to model count data, due to its dependency assumption.

This thesis is constructed around six related manuscripts, in which we propose several approaches for high-dimensional sparse count data clustering via various mixture models based on

hierarchical Bayesian modeling frameworks that have the ability to model the dependency of repetitive word occurrences. In such frameworks, a suitable distribution is used to introduce the prior information into the construction of the statistical model, based on a conjugate distribution to the multinomial, *e.g.* the Dirichlet, generalized Dirichlet, and the Beta-Liouville, which has numerous computational advantages. Thus, we proposed a novel model that we call the Multinomial Scaled Dirichlet (MSD) based on using the scaled Dirichlet as a prior to the multinomial to allow more modeling flexibility. Although these frameworks can model burstiness and overdispersion well, they share similar disadvantages making their estimation procedure is very inefficient when the collection size is large. To handle high-dimensionality, we considered two approaches. First, we derived close approximations to the distributions in a hierarchical structure to bring them to the exponential-family form aiming to combine the flexibility and efficiency of these models with the desirable statistical and computational properties of the exponential family of distributions, including sufficiency, which reduce the complexity and computational efforts especially for sparse and high-dimensional data. Second, we proposed a model-based unsupervised feature selection approach for count data to overcome several issues that may be caused by the high dimensionality of the feature space, such as over-fitting, low efficiency, and poor performance.

Furthermore, we handled two significant aspects of mixture based clustering methods, namely, parameters estimation and performing model selection. We considered the Expectation-Maximization (EM) algorithm, which is a broadly applicable iterative algorithm for estimating the mixture model parameters, with incorporating several techniques to avoid its initialization dependency and poor local maxima. For model selection, we investigated different approaches to find the optimal number of components based on the Minimum Message Length (MML) philosophy. The effectiveness of our approaches is evaluated using challenging real-life applications, such as sentiment analysis, hate speech detection on Twitter, topic novelty detection, human interaction recognition in films and TV shows, facial expression recognition, face identification, and age estimation.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my extraordinary supervisor, Prof. Nizar Bouguila, for his guidance, encouragement, and unconditional availability whenever needed not only for scientific matters but also for life. I will always be grateful to you for believing in me, for everything I have learned from you and for every single opportunity you have afforded. Thank you for your patience, caring, understanding, and continuous support. You have made a significant impact in my life to be the person and scientist who I am today. I am incredibly proud to have the honor of working under your supervision.

I would also like to thank my committee for their valuable time in reviewing my work, and for their careful, constructive and insightful comments that have significantly helped me to improve my work and widen my research horizon. I am forever grateful to my country, The Kingdom of Saudi Arabia, for the financial support to pursue my graduate studies in Canada. Moreover, I will not forget to thank Concordia University, where I have spent around seven years during my Master's and Ph.D. Besides, I am grateful for the conference and exposition award that I have received from the School of Graduate Studies several times during my studies. I would also like to thank my former and current fellow lab-mates for the quality times and unforgettable memories. Special thanks go to Muhammad Azam for his helpful suggestions and discussions during my research, especially at the early stage of the program.

Last, but by no means least, I would like to thank my family and warm-hearted friends, I would have never done this without your love and support. I want to especially thank my sister from another mister, Reham Fadul, my best friend, who has always been there for me during this journey.

Contribution of Authors

This Ph.D. thesis consists of six manuscripts, the first three have been published, and the rest have been submitted for publication in refereed academic journals. Each chapter consists of the content of a manuscript that has been reformatted and reorganized according to the requirements set out in the guideline by the School of Graduate Studies.

Manuscript 1 (Chapter 2): Zamzami, N., and Bouguila, N. (2019). Model selection and application to high-dimensional count data clustering. *Applied Intelligence*, 49(4), 1467-1488.

Manuscript 2 (Chapter 3): Zamzami, N., and Bouguila, N. (2019). A Novel Scaled Dirichlet-based Statistical Framework for Count Data Modeling: Unsupervised Learning and Exponential Approximation. *Pattern Recognition*, 95, 36-47.

Manuscript 3 (Chapter 4): Zamzami, N., and Bouguila, N. (2019). Hybrid generative discriminative approaches based on Multinomial Scaled Dirichlet mixture models. *Applied Intelligence*, 49(11), 3783-3800.

Manuscript 4 (Chapter 5): Zamzami, N., and Bouguila, N. (2019). High-Dimensional Count Data Clustering Based on an Exponential Approximation to the Multinomial Beta-Liouville Distribution. *Information Sciences*, manuscript submitted for publication.

Manuscript 5 (Chapter 6): Zamzami, N., and Bouguila, N. (2019). Sparse Count Data Clustering Using an Exponential Approximation to Generalized Dirichlet Multinomial Distributions. *IEEE Transactions on Neural Networks and Learning Systems*, manuscript submitted for publication.

Manuscript 6 (Chapter 7): Zamzami, N., and Bouguila, N. (2019). A Novel MM Framework for Simultaneous Feature Selection and Clustering of High-Dimensional Count Data. *IEEE Transactions on Cybernetics*, manuscript submitted for publication.

Contents

List of Figures xii

List of Tables xv

1 Introduction 1

1.1 Contributions . . . 2

1.2 Thesis Overview . . . 4

2 Model Selection and Application to High-dimensional Count Data Clustering via Finite EDCM Mixture Models 7

2.1 Introduction . . . 7

2.2 Related Works . . . 9

2.3 Finite EDCM Mixture Model . . . 11

2.3.1 The Dirichlet Compound Multinomial (DCM) Distribution . . . 11

2.3.2 The Exponential-family Approximation to DCM (EDCM) . . . 12

2.3.3 EDCM Mixture Model Learning . . . 13

2.4 The MML Criterion for EDCM Mixture . . . 15

2.4.1 Fisher Information for a Mixture of EDCM Distributions . . . 16

2.4.2 Prior Distribution $h(\Theta)$ for EDCM . . . 17

2.4.3 Algorithm for EDCM Mixture Estimation and Selection . . . 18

2.5 Experimental Results . . . 19

2.5.1 Text Documents Modeling . . . 20

2.5.2 Topic Novelty Detection . . . 23

2.5.2.1 Online Learning Framework . . . 23

2.5.2.2 Data, Evaluation Metrics and Results . . . 26
2.5.3 Hierarchical Image Categorization . . . 29
2.5.3.1 Hierarchical Clustering Approach . . . 29
2.5.3.2 Data Representation and Results . . . 31
2.6 Conclusion . . . 36

3 A Novel Scaled Dirichlet-based Statistical Framework for Count Data Modeling: Unsupervised Learning and Exponential Approximation 40

3.1 Introduction . . . 41
3.2 Hierarchical Bayesian Models for Count Data . . . 43
3.2.1 Dirichlet Compound Mutinomial (DCM) . . . 43
3.2.2 Efficient Alternative Priors . . . 44
3.3 The Proposed Model . . . 44
3.3.1 Multinomial Scaled Dirichlet (MSD) . . . 45
3.3.2 The Multinomial Scaled Dirichlet Mixture Estimation . . . 46
3.4 Approximating the MSD . . . 49
3.4.1 An Exponential-family Approximation to MSD (EMSD) . . . 49
3.4.2 Mixture of EMSDs Learning . . . 50
3.5 MML Criterion for EMSD Mixture . . . 51
3.6 Experimental Results . . . 54
3.6.1 Text Classification . . . 54
3.6.2 Facial Expression Recognition . . . 57
3.6.3 Texton-based Texture Clustering . . . 60
3.7 Conclusion . . . 62

4 Hybrid Generative/Discriminative Approaches Based on Multinomial Scaled Dirichlet Mixture Models 67

4.1 Introduction . . . 68
4.2 Related Works . . . 69
4.2.1 Generative Models for Count Data . . . 69

4.2.2 The Generative/Discriminative Learning Approach . . . 71
4.3 Finite Multinomial Scaled Dirichlet mixture model . . . 73
4.3.1 Multinomial Scaled Dirichlet (MSD) . . . 73
4.3.2 MSD Mixture Learning . . . 74
4.3.3 MSD Approximation to the Exponential Family . . . 77
4.4 A Hybrid of MSD/EMSD Mixture Models and SVM . . . 78
4.4.1 Development of Fisher Kernels . . . 79
4.4.2 Kernels Based on Information Divergence . . . 80
4.5 Experimental results . . . 83
4.5.1 Object Categorization . . . 83
4.5.1.1 Comparison of Generative and Discriminative Approaches . . . 84
4.5.1.2 Classification Results Using the Hybrid Approach . . . 85
4.5.2 Visual Scenes Modeling and Classification . . . 87
4.6 Conclusion . . . 90

5 High-Dimensional Count Data Clustering Based on an Exponential Approximation to the Multinomial Beta-Liouville Distribution 94

5.1 Introduction . . . 95
5.1.1 Motivations . . . 95
5.1.2 Contributions . . . 97
5.1.3 Organization . . . 97
5.2 Related Works . . . 98
5.3 The proposed Model . . . 99
5.3.1 Multinomial Beta-Liouville (MBL) Distribution . . . 99
5.3.2 The Exponential Multinomial Beta-Liouville (EMBL) . . . 100
5.3.3 The Learning Approach for EMBL Mixture Model . . . 102
5.3.3.1 Estimating the Number of Components . . . 102
5.3.3.2 The Component-wise Expectation Maximization (CEM) . . . 104
5.3.3.3 The Complete Algorithm for EMBL Mixture Model Learning . . . 105

5.3.4 Perspectives on the Proposed Model Efficiency . . . 107
5.4 Experimental Results . . . 108
5.4.1 Sentiment Analysis . . . 109
5.4.2 Shape Clustering Using Shape Context . . . 112
5.4.3 Recognition of Human Interactions in Films and TV Shows . . . 114
5.4.4 Distinguishing Male and Female Faces Using Generative Kernels . . . 117
5.4.4.1 Learning Approach and Datasets . . . 117
5.4.4.2 Generative Models Validation via BOF Approach . . . 121
5.4.4.3 Classification Results Using Generative/Discriminative Approach 122
5.5 Conclusion . . . 124

6 Sparse Count Data Clustering Using an Exponential Approximation to Generalized Dirichlet Multinomial Distributions 129

6.1 Introduction . . . 130
6.2 Related Works and Motivation . . . 132
6.3 The Generalized Dirichlet Multinomial Distribution . . . 133
6.4 Exponential-Family Approximation to GDM . . . 135
6.4.1 The Exponential Family of Distributions . . . 135
6.4.2 Approximating the GDM . . . 136
6.5 Estimation and Selection for a Finite mixture of EGDMs . . . 137
6.5.1 Maximum Likelihood Estimation . . . 138
6.5.2 MML Criterion for EGDM . . . 140
6.5.2.1 Fisher Information . . . 140
6.5.2.2 Prior Distribution . . . 142
6.6 Experimental Results . . . 144
6.6.1 Text Documents Modeling . . . 144
6.6.2 Image Database Categorization . . . 146
6.6.3 Human Action Recognition . . . 150
6.6.4 Model Selection Evaluation . . . 153

6.7 Conclusion 155

7 A Novel MM Framework for Simultaneous Feature Selection and Clustering of High-Dimensional Count Data **160**

7.1 Introduction 161

7.2 Related Works 163

7.3 The Proposed Model 165

7.3.1 An Alternative Representation for Generalized Dirichlet Multinomial (GDM) 166

7.3.2 Mixture Model with Feature Saliency 167

7.4 Model Learning 169

7.4.1 Parameters Estimation 169

7.4.2 Model Selection 172

7.4.3 The Complete Unsupervised Feature Saliency Algorithm 173

7.5 Experimental Results 175

7.5.1 Hate Speech and Offensive Language Detection on Twitter 175

7.5.2 Face Identification 178

7.5.3 Race Recognition 179

7.5.4 Age Estimation 181

7.6 Conclusion 183

8 Conclusion **186**

Bibliography **189**

List of Figures

Figure 2.1 Values of the different model selection criteria for the text datasets. 24
Figure 2.2 Overview of the proposed online framework for topic novelty detection. . . 25
Figure 2.3 Performance vs. fraction of documents in Online TDT framework based on EDCM mixture model. 28
Figure 2.4 Message length values as a function of the number of clusters for the whole data $\mathcal{X}_{N+1}$ in TDT2 dataset. 29
Figure 2.5 Message length values as a function of the number of clusters for the whole data $\mathcal{X}_{N+1}$ in 20Newsgroups dataset. 30
Figure 2.6 The graphical representation of the hierarchical clustering resulting in a tree-structured graph(dendrogram). 31
Figure 2.7 Example images of the PPMI dataset. 32
Figure 2.8 Number of clusters found by the different criteria for the different image datasets. 33
Figure 2.9 Confusion matrix for clustering the CIFAR-10 dataset using an EDCM mixture model. 34
Figure 2.10 Dendrogram showing the closest categories in CIFAR-10 based on KL divergence. 35
Figure 2.11 (a) Example images of the 13 categories Natural Scene dataset (b) Average accuracy of each level of the tree after merging similar clusters based on KL divergence. 36
Figure 2.12 (a) Confusion matrix for the PPMI (b) Dendrogram showing the closest categories based on KL divergence. 37

Figure 3.1 Number of clusters found by the MML criterion for the text data sets (a)IMBD, (b)Reuters-10, (c)WebKB4, and (d)20newsgroups. 56
Figure 3.2 Sample facial expression images from the MMI dataset. 58
Figure 3.3 Sample facial expression images from the CK+ dataset. 58
Figure 3.4 Confusion matrix for the six categories in the CK+ dataset using MSD (left), and EMSD (right). 59
Figure 3.5 Message length values as a function of the clusters number for the facial expression datasets. 60
Figure 3.6 Examples of images from the texture datasets. 61
Figure 4.1 Hierarchical representation of MSD model. 74
Figure 4.2 Graphical representation of the proposed hybrid learning approach. 79
Figure 4.3 Samples from Caltech and ETHZ datasets. 84
Figure 4.4 Samples from Fruits-360 dataset. 84
Figure 4.5 Sample images from the first dataset by Fei-Fei and Perona. 88
Figure 4.6 Sample images from MIT places dataset; Row1: Outdoor images, Row2: Indoor images. 88
Figure 5.1 Visualizing the performance of the proposed algorithm (EMBL) and original MBL against different input sizes considering memory usage (top row), and run time (bottom row) for (a) text dataset (IMDB: $N = 25,000$ $K = 2$), (b) shape dataset (Swedish Leafs $N = 600$ $K = 15$), (c) video dataset (High Five $N = 100$ $K = 4$), and (d) image dataset (Caltech Faces $N = 200$ $K = 2$). 107
Figure 5.2 (a) MPEG7CE-1 Set B dataset representative shapes,(b) Leaf dataset representative shapes. 114
Figure 5.3 Average and Intra-class accuracy for both actions in Kiss and Slap dataset. . 115
Figure 5.4 High Five dataset snapshots with different scale and camera views. 116
Figure 5.5 Samples from the face recognition datasets. 118
Figure 5.6 Clustering performance obtained for the different datasets using different techniques considering the BOF approach in different face recognition datasets. . . 122

Figure 6.1 (a) Sample images from CIFAR-10 dataset. (b) Intra-class accuracy obtained by GDM vs. EGDM for CIFAR-10. 148
Figure 6.2 (a) Sample images from SUN dataset. (b) Confusion matrix for SUN using EGDM. 149
Figure 6.3 Sample frames from Ballet video dataset. 151
Figure 6.4 Sample frames from YouTube Action dataset. 153
Figure 6.5 (a) Comparison of clustering performance for YouTube dataset using EDCM and EGDM, the average accuracy are 80.40% and 83.24%, respectively. (b) The confusion matrix for clustering using the proposed EGDM. 153
Figure 6.6 Number of clusters found by the MML criterion for the different datasets. . 154
Figure 7.1 Confusion matrices for detecting hate speech and offensive language using different approaches. 177
Figure 7.2 Sample images from the considered classes in PublicFig+LFW dataset. . . . 178
Figure 7.3 Sample images from UTK faces dataset. 180
Figure 7.4 Confusion matrix obtained by GDM with FS for race recognition in the UTK face dataset. 181

List of Tables

Table 2.1 Summarized text datasets properties (N: number of documents, $\overline{n}_d$: average document length, W: vocabulary size, M: true number of classes) 21

Table 2.2 Clustering results of text datasets (average $\pm$ slandered error) using a mixture of EDCMs. 22

Table 2.3 Summary of datasets characteristics (N: number of documents, $\overline{n}_d$: average document length, W: vocabulary size, M: true number of classes) 27

Table 2.4 Performance of different models comparing to online EDCM framework (NMI: normalized mutual information, $F1$: F score micro-averaged, $cdet$: cost of detect). 28

Table 2.5 Comparison of image databases categorization results using different clustering approaches. 33

Table 3.1 Description of the text data sets and comparison of the running time for MSD and EMSD (N:number of documents, W: vocabulary size, $\overline{n_i}$: average document length, M: number of classes). 55

Table 3.2 Classification results for IMDB dataset using MSD/EMSD mixture models. . 56

Table 3.3 Classification results for Reuters-10 dataset using MSD/EMSD mixture models. 56

Table 3.4 Classification results for WebKB4 dataset using MSD/EMSD mixture models. 57

Table 3.5 Classification results for 20newsgroups dataset using MSD/EMSD mixture models. 57

Table 3.6 Facial expression recognition results (average %) using MSD/EMSD mixture models. 59

Table 3.7 Texture image datasets considered in the experiments. 61

Table 3.8 Texture classification accuracy using different approaches. 62

Table 4.1 Object categorization performance obtained for the different datasets using different techniques considering the BOF approach. 85
Table 4.2 Object categorization performance comparison for the hybrid learning using different kernels. 86
Table 4.3 Object categorization performance obtained by fitting directly different generative models to the local descriptors. 86
Table 4.4 Visual scenes classification performance obtained for the different visual scenes datasets using different techniques considering the BOF approach. 88
Table 4.5 Visual scenes classification performance comparison using different kernels. 89
Table 4.6 Visual scenes classification performance obtained by fitting directly different generative models to the local descriptors. 89
Table 5.1 Clustering results for the IMDB dataset using EMBL mixture. 110
Table 5.2 Clustering results for the Amazon dataset using EMBL mixture. 111
Table 5.3 Clustering results for the Yelp dataset using EMBL mixture. 111
Table 5.4 Comparison of our method to the best published results (avg accuracy %) from previous works for sentiment analysis. 112
Table 5.5 Shape clustering performance (avg. accuracy %) using different generative models. 113
Table 5.6 Comparison of our method with the state of the art for the Swedish leaf database. 114
Table 5.7 Average precision results and time for recognizing human interaction in High Five dataset using different generative models 116
Table 5.8 Comparison of our method with the state of the art for High Five dataset. . . 116
Table 5.9 Classification performance obtained for the different face datasets using different techniques considering the BOF approach. 121
Table 5.10 Performance (%) for gender faces distinguishing comparison of different generative kernels. 123
Table 5.11 Clustering performance by fitting directly the different generative models to faces datasets. 123
Table 5.12 Comparison of our method with the state of the art for the AR face dataset. . 124

Table 6.1 Clustering results using EGDM mixture model for the three documents collections. 147
Table 6.2 The average accuracy and learning time using different methods for image categorization. 149
Table 6.3 The average accuracy and learning time using different methods for human action recognition. 151
Table 7.1 Clustering results (%) for Twitter dataset over 20 random runs. 176
Table 7.2 Average accuracy (%) per-class for face identification over 20 random runs. . 179
Table 7.3 Average clustering results (%) for race recognition over 20 random runs. . . 181
Table 7.4 Average clustering results (%) for age estimation over 20 random runs. . . . 182

Chapter 1

Introduction

Clustering, the process of discovering the natural grouping of a set of objects and assigning observations sharing similar characteristics to subgroups, is a significant task in data analysis and pattern recognition that has attracted considerable attention of scholars in the last decades. Numerous scientific fields and applications have utilized clustering techniques with different algorithms. Statistical-based approaches are robust and widely used in generative learning processes to abstract the complexity of a vast amount of information. One primary approach is finite mixture models that permit a formal technique for unsupervised learning [1]. Mixture models are used to model data sampled from a finite number of homogeneous subpopulations, where the whole model is formed by a weighted sum of the subgroups densities [2, 3]. Due to their flexibility, mixture models are adopted in many applications related, for instance, to image processing and computer vision [4], social networks [5], and recommending systems [6]. Most of the clustering methods have been developed for the case of continuous data. However, count data naturally appear in numerous fields with several applications in machine learning and computer vision. Consider, for instance, image categorization and other computer vision tasks, sentiment analysis, and documents that can be clustered to generate topical hierarchies for efficient information access or retrieval. Clustering count data is a challenging task due to its high-dimensionality and sparse nature. Furthermore, count data are usually characterized by burstiness and overdispersion phenomena [7, 8].

Hierarchical Bayesian modeling frameworks, such as Dirichlet Compound Multinomial (DCM) [9], Generalized Dirichlet Multinomial (GDM) [10] and Multinomial Beta-Liouville (MBL) [11]

have shown to be competitive with the best-known clustering methods for count data that address these phenomena and outperform the widely used multinomial model. However, their estimation procedures are very inefficient when the collection size is large. Indeed, processing high-dimensional data requires significantly increasing time and space. Thus, to handle high-dimensional data, an efficient exponential-family approximation to the DCM (EDCM) has been previously proposed by Elkan [12]. Exponential family of distributions has finite-sized sufficient statistics, meaning that we can compress the data into a fixed-sized summary without loss of information [13, 14]. EDCM has shown to address the burstiness phenomenon successfully and to be considerably computationally faster than DCM, especially when dealing with sparse and high-dimensional vectors. Another approach to handle high dimensionality is feature selection, which aims at finding the most relevant feature subset from high-dimensional feature space based on certain evaluation criteria [15–17]. Thus, feature selection helps in improving the statistical model structure and overcoming several issues that may be caused by the high dimensionality of the feature space such as over-fitting, low efficiency, and poor performance [18–20]. This thesis is motivated by the growing demand to handle high-dimensional and sparse frequency vectors that appear in many real-life applications.

1.1 Contributions

The goal of this thesis is to propose several novel approaches for high-dimensional sparse count data clustering and classification based on various mixture models frameworks. The contributions of this thesis are listed as the following:

- ☞ **An MML Criterion to Determine the Number of Components in EDCM Mixture**

  we develop an MML criterion to determine the number of components in EDCM mixture as an efficient unsupervised learning algorithm for clustering high-dimensional and sparse count data. This work is an extended version of our earlier work [21], as we further extend the proposed approach to different challenging count data clustering tasks. Based on the EDCM mixture and the MML criterion, we proposed a probabilistic model for online document clustering with application to the topic novelty detection, which can be viewed as one of the significant contributions of this paper. The other contribution concerns proposing a

distance-based agglomerative clustering approach for hierarchical image categorization using a mixture of EDCMs. This work is published in *Applied Intelligence* journal [22].

☞ **A Novel Scaled Dirichlet-based Statistical Framework for Count Data Modeling**

we propose a novel statistical framework, which is the composition of the scaled Dirichlet distribution and the multinomial. We initially proposed the model in [23], and we called it the Multinomial Scaled-Dirichlet (MSD). In addition, we derive a new distribution that is a close approximation to the MSD as a member of the exponential family of distributions that we called (EMSD). Then, for determining the number of components in the EMSD mixture, we develop a Minimum Message Length (MML) criterion. By means of some challenging applications, we show that both MSD and EMSD are better suited than the multinomial and DCM for modeling count data. This work is published in *Pattern Recognition* journal [24].

☞ **Hybrid Generative/Discriminative Approaches Based on Mixture Models**

we propose a hybrid model devoted to the applications in which count data representations are involved. Several well-motivated SVM kernels have been developed based on MSD/EMSD mixture models. In particular, we develop a Fisher kernel between two MSD/EMSD distributions and closed-form expressions of different information-divergence based kernels, namely, Kullback–Leibler kernel, Reńyi kernel, and Jensen–Shannon kernel. This work is published in *Applied Intelligence* journal [25].

☞ **Approximating MBL as a Member of the Exponential-Family for High-Dimensional Count Data Clustering**

we propose an exponential approximation to the Multinomial Beta-Liouville (MBL) that improves its performance and computation complexity. Moreover, we propose a learning approach that is robust in terms of initialization and simultaneously deals with fitting the mixture model to the observed data and selecting the optimal number of components, which makes it efficient for large datasets. Furthermore, we build new probabilistic kernels based on information divergences and Fisher scores from the proposed mixture of EMBL for Support Vector Machines (SVMs) as a powerful hybrid learning approach. This work in under review by the *Information Sciences* journal [26].

☞ **An Exponential Approximation to Generalized Dirichlet Multinomial Distributions**

we derive a new distribution that is a close approximation to the GDM as a member of the exponential family of distributions that we called EGDM. Furthermore, we developed a clustering framework via a mixture of EGDMs. For learning the parameters of an EGDM mixture, we propose the use of the Deterministic Annealing Expectation-Maximization (DAEM) algorithm to avoid the initialization dependency problem of the standard EM. Based on the EGDM mixture and the MML criterion, we proposed a probabilistic model for different challenging clustering tasks that involve high-dimensional sparse count data. This work is under review by *IEEE Transactions on Neural Networks and Learning Systems* [27].

☞ **A Novel MM Framework for Simultaneous Feature Selection and Clustering of High-Dimensional Count Data**

we propose a probabilistic feature selection approach that considers discrete random variables modeled by a finite mixture of GDM distributions. Moreover, we derive a minorization-maximization (MM) algorithm for estimating the proposed mixture with feature saliencies where the surrogate function is much simpler than the log-likelihood, and thus the M step can be solved analytically. For clustering, we propose an unsupervised learning approach that simultaneously deals with fitting the mixture model to the observed data and selecting the optimal number of components. We validate the proposed model via challenging clustering problems that involve multimedia data with high-dimensional discrete feature spaces. This work is under review by *IEEE Transactions on Cybernetics* [28].

1.2 Thesis Overview

The organization of this thesis is as follows:

- Chapter 1 introduces the background knowledge regarding finite mixture models for count data and provides an overview of the thesis.
- In Chapter 2, we propose the use of Minimum Message Length (MML) criterion for determining the number of components that best describes the data with a finite EDCM mixture

model. Parameters estimation is based on the previously proposed Deterministic Annealing Expectation- Maximization (DAEM) approach. The validation of the proposed unsupervised algorithm involves different real applications: text document modeling, topic novelty detection, and hierarchical image classification.

- In Chapter 3, we propose a novel model called the Multinomial Scaled Dirichlet (MSD) distribution that is the composition of the scaled Dirichlet distribution and the multinomial. Moreover, to improve the computation efficiency in high-dimensional spaces, we propose to approximate the MSD as a member of the exponential family. The performance evaluation of the proposed models is conducted through a set of extensive empirical experiments on challenging applications, namely; text classification, facial expression recognition, and texture images clustering.

- In Chapter 4, we combine the advantages and desirable properties of generative models, *i.e.,* finite mixture, and the Support Vector Machines (SVMs) as powerful discriminative techniques for modeling count data. In particular, we select accurate kernels generated from mixtures of Multinomial Scaled Dirichlet distribution and its exponential approximation (EMSD) for support vector machines. We demonstrate the effectiveness and the merits of the proposed framework through challenging real-world applications, namely; object recognition and visual scenes classification.

- In Chapter 5, we propose a mixture model for high-dimensional count data clustering based on an exponential-family approximation of the Multinomial Beta-Liouville distribution, which we call EMBL. We deal simultaneously with the problems of fitting the model to observed data and selecting the number of components. The learning algorithm automatically selects the optimal number of components and avoids several drawbacks of the standard Expectation-Maximization algorithm, including the sensitivity to initialization and possible convergence to the boundary of the parameter space. We demonstrate the effectiveness and robustness of the proposed clustering approach through a set of extensive empirical experiments that involve challenging real-world applications such as sentiment analysis, shape categorization,

and human iterations in movies and TV shows. Moreover, we proposed a hybrid generative/discriminative learning approach based on the mixture of MBL/EMBL and validated it for distinguishing male and female faces.

- In Chapter 6, we derive a new family of distributions that approximates the GDM distributions, and we call it (EGDM). A mixture model is developed based on the new exponential family of distributions, and its parameters are learned through the Deterministic Annealing Expectation-Maximization (DAEM) approach as a new clustering algorithm for count data. Moreover, we propose the use of the Minimum Message Length (MML) criterion for selecting the optimal number of components to describe the data with a finite EGDM mixture model best. A set of empirical experiments, which concern text documents modeling, natural scenes categorization, and human action recognition, have been conducted to evaluate the proposed approach performance.

- In Chapter 7, we propose a probabilistic approach for count data based on the concept of feature saliency in the context of mixture-based clustering using the generalized Dirichlet multinomial (GDM) distribution. By minimizing the message length, the saliency of irrelevant features is driven toward zero, which corresponds to performing feature and model selection simultaneously. Through a set of challenging applications, it is demonstrated that the developed approach performs effectively in selecting both the optimal number of clusters and the most relevant features and, thus, improve the clustering performance considerably in different real-world applications including hate speech detection, face identification, race recognition, and age estimation.

- In Chapter 8, we conclude the thesis by highlighting the main findings, summarizing our contributions, and presenting some promising future research directions.

Chapter 2

Model Selection and Application to High-dimensional Count Data Clustering via Finite EDCM Mixture Models

EDCM, the Exponential-family approximation to the Dirichlet Compound Multinomial (DCM), proposed by Elkan [12], is an efficient statistical model for high-dimensional and sparse count data. EDCM models take into account the burstiness phenomenon correctly while being many times faster than DCM. This work proposes the use of the Minimum Message Length (MML) criterion for determining the number of components that best describes the data with a finite EDCM mixture model. Parameters estimation is based on the previously proposed Deterministic Annealing Expectation-Maximization (DAEM) approach. The validation of the proposed unsupervised algorithm involves different real applications: text document modeling, topic novelty detection, and hierarchical image classification. A comparison with results obtained for other information-theory based selection criteria is provided.

2.1 Introduction

In data analysis and pattern recognition, a challenging task is clustering, the process of discovering the natural grouping of a set of objects, and assigning observations sharing similar characteristics

to subgroups [29]. Numerous scientific fields and applications have utilized clustering techniques with different algorithms. Consider for instance, image segmentation and other computer vision tasks [30, 31], documents that can be clustered to generate topical hierarchies for efficient information access [32, 33], or retrieval [34]. For applications where time plays an increasingly essential role, treating the new coming data as soon as it arrives in a temporal sequence is an important issue to satisfy the users' needs [35, 36]. Online clustering, also called incremental clustering, is a demanding unsupervised learning task that needs to be done in an online setting. Topic detection and tracking [37, 38], dynamic image databases summarization [36], and object recognition in video [39], are few examples for online clustering applications.

Statistical-based approaches are powerful and widely used in generative learning processes to abstract the complexity of a huge amount of information. One major approach based on statistics is finite mixture models that permit a formal approach for unsupervised learning [1]. They are used to model data sampled from a finite number of homogeneous subpopulations, where the whole model is formed by a weighted sum of the subgroups densities [2, 3]. Due to their flexibility, mixture models are adopted in many applications, including, but not limited to, image processing and computer vision [4], social networks [5], and recommending systems [6]. An essential issue in mixture modeling is selecting the optimal number of components that best describes and represent the data [40, 41]. For instance, given a set of documents, users have to browse the whole document collection in order to estimate the number of topics K, which is not only time consuming but also unrealistic, especially when dealing with large datasets. Furthermore, an improper estimation of K might easily mislead the clustering process as using a bigger or a smaller number of clusters ultimately degrades clustering accuracy [42].

On the other hand, classes that are represented by different modes in the mixture are indeed generalizations of each other and can be considered as sub-clusters of the main cluster in many domains [35, 43]. Thus, hierarchical clustering is one of the most frequently used schemes in unsupervised learning to represent the underlying application domain naturally. Given a set of data points that are sampled from a mixture of distributions, the clustering output is a binary tree (dendrogram) that organizes the clusters hierarchically, where this hierarchy agrees with the intuitive organization of real-world data [44, 45]. Hierarchical trees provide a view of the data at different

levels of abstraction, which allows flat partitions of different granularity to be extracted during data analysis, making them ideal for interactive exploration and visualization [45]. The problem of hierarchical mixture modeling has been addressed in the literature either in divisive (top-down) mode [46, 47], or agglomerative (bottom-up) mode [48] based on probabilistic distances [49, 50], or interclass correlations [51].

In this paper, we develop an MML criterion to determine the number of components in the EDCM mixture as an efficient unsupervised learning algorithm for clustering high-dimensional and sparse count data. This paper is an extended version of our earlier work [21], in which we evaluated the efficiency of the proposed approach in determining the number of topics within a document collection. Here, we further extend the work to different challenging clustering tasks that involve high dimensional count data. Based on the EDCM mixture and the MML criterion, we propose a probabilistic model for online document clustering with application to topic novelty detection, which can be viewed as one of the major contributions of this paper. The other contribution concerns proposing a distance-based agglomerative clustering approach for hierarchical image classification using a mixture of EDCMs.

The rest of the article is organized as follows. Section 2.2 discusses previous relevant works. In Section 3, we review the EDCM mixture model and the estimation of its parameters. In Section 2.3, we review EDCM mixture model and its parameters estimation. The MML expression for the EDCM mixture, as well as the complete algorithm for estimation and selection, are detailed in Section 2.4. Section 2.5 presents and discusses the clustering applications experiments and results. Section 2.6 ends the paper with some concluding remarks.

2.2 Related Works

Clustering methods can be categorized based on whether the number of clusters is required as the input parameter. If the number of clusters is predefined, many algorithms based on the probabilistic finite mixture model have been provided in the literature, including the multinomial mixture model that applies the EM algorithm for document clustering, for instance, assuming that document topics follow multinomial distribution [52]. Deterministic annealing procedures are proposed

to allow this algorithm to find better local optima of the likelihood function [53]. The multinomial distribution is often used to model text document, and it makes a naive Bayes independence assumption, *i.e.,* each word of the document is generated independently from the others, which is not valid for word emissions in the natural text where words tend to appear in bursts [7, 54]. An alternative approach for modeling term frequencies is hierarchical Bayesian modeling that introduces the Dirichlet distribution as a prior to the Multinomial, which results in Dirichlet Compound Multinomial (DCM) [9]. The hierarchical approach of DCM considers the count vector to be generated by a multinomial distribution whose parameters are generated by the Dirichlet distribution. This composition that is based mainly on the fact that the Dirichlet is a conjugate to the multinomial offers numerous computational advantages [55]. The experiment by Madsen et al. [9] showed that the performance of DCM was comparable to that obtained with multiple heuristic changes to the multinomial model. However, the DCM model lacks intuitiveness, and the parameters in that model cannot be estimated quickly. Elkan [12] derived the EDCM distribution, which belongs to the exponential family, which is a good approximation to the DCM distribution. The EM algorithm with the EDCM distributions has shown to attain high clustering accuracy while being much faster than the corresponding algorithm with DCM distributions proposed in [9]. In recent years, the EM algorithm with EDCM distribution is the most competitive algorithm for document clustering in case the number of clusters is predefined.

If the number of clusters K is unknown before the clustering process, one solution is to estimate K first and use this estimation as the input parameter for those document clustering algorithms requiring K predefined. Several other approaches have been proposed in the literature to find the optimal number of clusters K. The most straightforward method is the likelihood cross-validation technique, which trains the model with different values of K and picks the one with the highest likelihood. Another approach is to assign a prior to K and then calculate the posterior distribution of K to determine its value, where the methods can be generally classified, from a computational point of view, into deterministic and stochastic methods. Since the stochastic schemes are computationally demanding, the majority of the used approaches are deterministic [3]. Examples of deterministic methods based on information/coding theory concept include Minimum Message Length (MML) [1, 56], Akaike's Information Criterion (AIC) [57], Minimum Description Length (MDL) [58],

Mixture MDL (MMDL) [1]. MML is a statistically consistent and efficient technique and its implementation as a model selection criterion has shown to give good results with mixtures models (for instance; with mixture of Gaussians [40], Poisson and von Mises circular distributions [41], and recently with mixture of Dirichlet distributions [59] and mixture of generalized Dirichlet distributions [60]).

2.3 Finite EDCM Mixture Model

In this section, we shall first summarize the hierarchical Bayesian model called Dirichlet Compound Multinomial (DCM), which is a composition of the multinomial and Dirichlet distributions. Moreover, we discuss the close approximation to the DCM that has been derived as a member of the exponential family of distributions by Elkan [12] called EDCM. Then, we present the deterministic annealing expectation-maximization algorithm for learning a mixture of EDCMs.

2.3.1 The Dirichlet Compound Multinomial (DCM) Distribution

Define $\mathbf{X} = (x_1, \ldots, x_W)$ as a sparse vector of counts representing a document, or an image, where x_w is the frequency of the word, or visual word, w. Then, the probability of $\mathbf{X}$ that it follows a multinomial distribution with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_W)$, is given by:

$$\mathcal{M}(\mathbf{X}|\boldsymbol{\alpha}) = \frac{n!}{\prod_{w=1}^{W} x_w!} \prod_{w=1}^{W} \alpha_w^{x_w} \tag{2.1}$$

where W is the size of the vocabulary, and $n = \sum_{w=1}^{W} x_w$.

Several limitations and technical problems associated with the multinomial assumption have been discussed in the literature [9, 61]. Consider, for example, the in-dependency assumption, *i.e.,* each word of the document is generated independently from every other, which is not valid for word emissions in natural text with burstiness phenomenon, once a word appears in a document it is much more likely to appear again [7, 54]. An appropriate and efficient solution to address this issue is the hierarchical Bayesian modeling approach that introduces the prior information into the construction of the statistical model. Generally, the natural conjugate prior to the multinomial assumption is the

Dirichlet distribution with a set of parameters $\varphi = (\varphi_1, \ldots, \varphi_W)$, defined as [62]:

$$\mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\varphi}) = \frac{\Gamma(s)}{\prod_{w=1}^{W} \Gamma(\varphi_w)} \prod_{w=1}^{W} \alpha_w^{\varphi_w - 1} \tag{2.2}$$

where $s = \sum_{w=1}^{W} \varphi_w$ is the sum of the parameters. Then, the Dirichlet Compound Multinomial (DCM) is the marginal distribution given by the integration over all possible Multinomials [9]:

$$\begin{aligned} \mathcal{DCM}(\mathbf{X}|\boldsymbol{\varphi}) &= \int_{\alpha} \mathcal{M}(\mathbf{X}|\boldsymbol{\alpha}) \mathcal{D}(\boldsymbol{\alpha}|\boldsymbol{\varphi}) d\alpha \\ &= \frac{n!}{\prod_{w=1}^{W} (x_w)!} \frac{\Gamma(s)}{\Gamma(\sum_{w=1}^{W} x_w + \varphi_w)} \prod_{w=1}^{W} \frac{\Gamma(x_w + \varphi_w)}{\Gamma(\varphi_w)} \end{aligned} \tag{2.3}$$

where $n = \sum_{w=1}^{W} x_w$ is the document length.

We can note that compared to the multinomial, the DCM has one extra degree of freedom, since its parameters are not restricted to satisfy the unit-sum constraint, which makes it more practical [61, 63]. However, DCM parameters cannot be estimated quickly in high-dimensional spaces [64].

2.3.2 The Exponential-family Approximation to DCM (EDCM)

Considering the sparsity nature of the datasets represented as bag-of-words, or bag-of-visual-words, a close approximation to the DCM has been derived as a member of the exponential family of distributions by Elkan that was called EDCM [12]. In such approximation, only non-zero word counts x_w are used for computation efficiency, given that most words do not appear in most documents. That is, we retain only the sufficient statistic for the purpose of estimating the parameters [65]. The EDCM, as an approximation to DCM, can be written as [12]:

$$\mathcal{EDCM}(\mathbf{X}|\varphi) = n! \frac{\Gamma(s)}{\Gamma(s+n)} \prod_{w:x_w \geq 1} \frac{\varphi_w}{x_w} \tag{2.4}$$

This form makes it clear how EDCM allows multiple appearances of the same word to have high probability. That is, the first appearance of a word w reduces the probability of a document by $\varphi_w \ll 1$, while mth appearance of any word reduces the probability by $(m-1)/m$ which tends

to 1 as m increases. Moreover, it distinguishes between word types and word tokens, as modeling both frequencies is beneficial for capturing the statistical properties of natural languages [66].

As the EDCM distribution is a member of the exponential family of distributions, its density can be written in the following form [14, 67]:

$$q(\mathbf{X}|\theta) = H(X)\Phi(\theta)\exp\{G(\theta).T(X)\} \tag{2.5}$$

where $T(X) = (T_1(X), \ldots, T_W(X))$ is a vector of sufficient statistics and $G(\theta)$ is the vector of natural parameters [65]. Thus, re-writing Eq.(2.4) as an exponential density gives:

$$q(\mathbf{X}|\varphi) = \left(\prod_{w:x_w\geq 1} x_w^{-1}\right) n! \frac{\Gamma(s)}{\Gamma(s+n)} \exp\left[\sum_{w=1}^{W} I(x_w \geq 1)\log\varphi_w\right]$$

where $I(x_w \geq 1)$ is an indicator equals to 1 if the word w appears at least once in a vector X, and 0 otherwise.

2.3.3 EDCM Mixture Model Learning

Given a documents collection $\mathcal{X}$ with N independent documents $\{\mathbf{X_1}, \ldots, \mathbf{X_N}\}$, an EDCM mixture with M components, different topics, is defined as:

$$P(\mathcal{X}|\Theta) = \sum_{j=1}^{M} \mathcal{EDCM}(\mathbf{X}|\boldsymbol{\varphi_j})\ \mu_j \tag{2.6}$$

where $(0 < \mu_j < 1$ and $\sum_{j=1}^{M}\mu_j = 1)$ are the mixing proportions. In this case, $\mathcal{X}$ represents a set of observed variables, and $\Theta = (\boldsymbol{\varphi_1}, \ldots, \boldsymbol{\varphi_M}, \mu_1, \ldots, \mu_M)$ denotes the set of all latent variables and parameters. In this case, the complete data are considered to be $\{\mathcal{X}, \mathcal{Z}|\Theta\}$, where $\mathcal{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_M\}$ denotes the missing group-indicator vectors for data elements in the jth cluster. The value of z_{ij} is equal to one if the observation $\mathbf{X}_i$ is generated by the cluster j and zero otherwise. Thus, the complete data log-likelihood corresponding to a M-component mixture is given by:

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{M} z_{ij}\ \log(P(\mathbf{X}_i|\boldsymbol{\varphi}_j)\ \mu_j) \tag{2.7}$$

Learning mixture models is possible through the Expectation Maximization (EM) algorithm [65, 68], which produces a sequence of estimates $\{\Theta^{(c)}, c = 1, 2, \dots\}$. For learning a mixture of EDCM distributions, Elkan [12] suggested using the deterministic annealing procedure that allows EM to avoid initialization dependency and poor local maxima [53]. Some interesting justifications about using the deterministic annealing procedure can be found in [12, 53].

In that procedure, three phases are considered where each phase runs EM until convergence. The temperature parameter T has been set to $T = 25, T = 5,$, and lastly $T = 1$, where the final Θ parameters in each phase are used as initial values in the next one. According to Elkan [12], slower annealing schedules provide no significant additional benefit in the case of EDCM. When applying the deterministic annealing procedure, the posterior probabilities will be computed in the **E-step** as:

$$z_{ij}^{(c)} = \frac{\left(P(\mathbf{X}_i|\boldsymbol{\varphi}_j^{(c)})\ \mu_j^{(c)}\right)^{\tau}}{\sum_{j=1}^{M}\left(P(\mathbf{X}_i|\boldsymbol{\varphi}_j^{(c)})\ \mu_j^{(c)}\right)^{\tau}} \tag{2.8}$$

where $\tau = \frac{1}{T}$. In the **M-step**, the parameters estimates will be updated according to:

$$\hat{\Theta}^{(c+1)} = \arg\max_{\Theta} \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta, \Theta^{(c)}, T) \tag{2.9}$$

when maximizing (2.9), we obtain:

$$\mu_j^{(c+1)} = \frac{1}{N}\sum_{i=1}^{N} z_{ij}^{(c)} \tag{2.10}$$

The weighted maximum-likelihood parameter vector in the case of EDCM can be obtained easily, as shown in [12]. Each φ_w for a component j can be computed using:

$$\varphi_{jw}^{(c+1)} = \frac{\sum_{i=1}^{N} z_{ij}^{(c)}\ I(x_{iw} \geq 1)}{\sum_{i=1}^{N} z_{ij}^{(c)}\ \Psi(s_j^{(c)} + n_i) - D\ \Psi(s_j^{(c)})} \tag{2.11}$$

where $D = \sum_{i=1}^{N} z_{ij}^{(c)}$, and Ψ is the digamma function.

2.4 The MML Criterion for EDCM Mixture

In this section, we determine the message length expression for the EDCM mixture and give the complete algorithm of estimation and selection. Minimum Message Length (MML) is an invariant point estimation that can be interpreted as it variously states that the best conclusion from data is the theory with the highest posterior probability [41]. Let D be the data and H be an hypothesis (or theory) with prior probability $Pr(H)$, MML information-theoretical interpretation is that an event of probability p can be coded by a message of length $l = -\log_2 p$ bits according to elementary coding theory. Hence, since we know that $-\log_2(Pr(H).Pr(D|H)) = -\log_2(Pr(H)) - \log_2(Pr(D|H))$, maximizing the posterior probability $Pr(H|D)$, is equivalent to minimizing the length of the two-part message [41, 69]:

$$Messlen = -\log_2(Pr(H)) - \log_2(Pr(D|H))$$

MML, as a model selection criterion, has shown to give good results with mixture modeling. From the information theory point of view, this selection criterion approach is based on evaluating statistical models according to their ability to compress a message containing the data, where high compression is obtained by forming good models of data to be coded [70]. For each model in the model space, the message includes two parts. The first part encodes the model using only prior information about the model and no information about the data. The second part encodes only the data in a way that makes use of the model encoded in the first part [69].

Let $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ a set of data controlled by a mixture of EDCM distributions with parameters $\Theta = (\theta_1, \ldots, \theta_M)$, where M is the number of clusters and θ_j is the parameter vector of the jth component. According to information theory, the optimal number of clusters M is the candidate value, which minimizes the amount of information, measured in $nats$ using the natural logarithm, to transmit $\mathcal{X}$ efficiently from a sender to a receiver [71]. For a mixture of distributions with N_p free parameters to be estimated, which is $M(W+1)-1$ in case of EDCM mixture, the

formula for the message length is given by [40, 41]:

$$\begin{aligned} MessLength \simeq & -\log(h(\Theta)) - \log(P(\mathcal{X}|\Theta)) \\ & + \frac{1}{2}\log(|F(\Theta)|) + \frac{N_p}{2}(1 + \log(k_{N_p})) \end{aligned} \tag{2.12}$$

where $h(\Theta)$ is the prior probability, $P(\mathcal{X}|\Theta)$ is the likelihood for the complete dataset, $|F(\Theta)|$ is the determinant of the expected Fisher information matrix, and kN_p is the optimal quantization lattice constant for $\mathbb{R}^{N_p}$ [72]. When $N_p = 1$ the value of $k_1 = 1/12 \simeq 0.083$, and as N_p grows, k_{N_p} tends to the asymptotic value given by $\frac{1}{2\pi e} \simeq 0.05855$ which can be approximated by $\frac{1}{12}$ [41].

The following sections show the detailed calculations for the determinant of Fisher information $|F(\Theta)|$, and the prior probability density function $h(\Theta)$ for a mixture of EDCM.

2.4.1 Fisher Information for a Mixture of EDCM Distributions

Fisher information is defined as the determinant of the Hessian matrix of minus the log-likelihood of the mixture [41]. In the case of the EDCM mixture, the complete-data Fisher information matrix has a block-diagonal structure. We assume that φ and μ are independent as the prior information about one would usually not be greatly influenced by the other. Moreover, the components of φ are also assumed to be independent. Thus, the complete-data Fisher information determinant is given as the product of the determinant of the Fisher information of $\boldsymbol{\varphi_j}$ for each component and the determinant of the Fisher information of mixing parameters vector $\boldsymbol{\mu}$ [1, 40], as follows:

$$|F(\Theta)| \simeq |F(\boldsymbol{\mu})| \prod_{j=1}^{M} |F(\boldsymbol{\varphi_j})| \tag{2.13}$$

Given that the mixing proportions satisfy the requirement $\sum_{j=1}^{M} \mu_j = 1$, it is possible to consider its Fisher information as a series of trails, and each has M possible outcomes. In this case, the number of trails of the jth cluster is a multinomial distribution with parameters $(\mu_1, \ldots, \mu_M)$.

Hence, the determinant of the Fisher information of the mixing parameters vector is [40]:

$$|F(\boldsymbol{\mu})| = \frac{N}{\prod_{j=1}^{M} \mu_j} \tag{2.14}$$

where N is the number of documents.

The Fisher information matrix in case of a mixture model can be computed after the data vectors have been assigned to their respective clusters, as proposed in [1]. Let $\mathcal{X}_j = \{\mathbf{X}_l, \ldots, \mathbf{X}_{l+\eta_j-1}\}$ be the data elements in the jth cluster where $l \leq N$ and η_j the number of the observations generated by the jth mixture component with parameters $\boldsymbol{\varphi}_j$. The negative of the log-likelihood function given the vector $\boldsymbol{\varphi_j} = (\varphi_1, \ldots, \varphi_W)$ and $s_j = \sum_{w=1}^{W} \varphi_{jw}$ of a single EDCM distribution is $-\mathcal{L}(\mathcal{X}_j|\boldsymbol{\varphi}_j)$. Thus, we can show that $\log(|F(\Theta)|)$ in case of finite EDCM mixture model is given by (see Appendix 1):

$$\begin{aligned}\log(|F(\Theta)|) \simeq \log(N) - \sum_{j=1}^{M} \log(\pi_j) + \sum_{j=1}^{M} \log\Bigg(|1 + \Bigg(\eta_j(-\Psi'(s_j)) + \sum_{d=l}^{l+\eta_j-1} \Psi'(s_j + n_d) \Bigg) \\ \times \sum_{w=1}^{W} \frac{1}{\sum_{d=l}^{l+\eta_j-1} I(x_{dw} \geq 1) \frac{1}{\varphi_{jw}^2}} | \Bigg) + \sum_{j=1}^{M} \sum_{w=1}^{W} \log\Big(\sum_{d=l}^{l+\eta_j-1} I(x_{dw} \geq 1) \frac{1}{\varphi_{jw}^2} \Big)\end{aligned} \tag{2.15}$$

2.4.2 Prior Distribution $h(\Theta)$ for EDCM

The capability of the MML criterion is controlled by the choice of prior distribution $h(\Theta)$ for the parameters of EDCM. In case of mixture models, we make a general assumption that the parameters of the different components as a prior are independent from the mixing probabilities, and the components of $h(\boldsymbol{\varphi}_j)$ are independent as well [73], that is:

$$h(\Theta) = h(\boldsymbol{\mu}) \prod_{j=1}^{M} h(\boldsymbol{\varphi}_j) = h(\boldsymbol{\mu}) \prod_{j=1}^{M} \prod_{w=1}^{W} h(\varphi_{jw}) \tag{2.16}$$

Knowing that the vector $\boldsymbol{\mu}$ is defined on the simplex $\{(\mu_1, \ldots, \mu_M) : \sum_{j=1}^{M} \mu_j = 1\}$, then the Dirichlet distribution is a natural choice as a prior for the mixing probabilities. The choice of a

constant Dirichlet parameters (a vector of ones) gives a uniform prior as follows [40, 41]:

$$h(\boldsymbol{\mu}) = \Gamma(M) = (M-1)! \tag{2.17}$$

For calculating $h(\boldsymbol{\varphi}_j)$ and in the absence of other knowledge about the $\varphi_{jw}, w = 1, \ldots, W$, we assume that $h(\varphi_{jw})$ is locally uniform over the range $[0, e^6 \frac{|\hat{\varphi}_j|}{\varphi_{jw}}]$ where $\hat{\varphi}_j$ is the estimated vector. We choose to use a simple uniform prior, which is known to give good results, in accordance with Ockham's razor [74], as:

$$h(\varphi_{jw}) = \frac{e^{-6}\varphi_{jw}}{|\hat{\boldsymbol{\varphi}}_j|} \tag{2.18}$$

Thus, substituting Eq.(2.18) and Eq.(2.17) into Eq.(2.16), and taking the log we obtain:

$$\log(h(\Theta)) = \sum_{j=1}^{M} \log(j) - 6MW - W \sum_{j=1}^{M} \log(|\hat{\boldsymbol{\varphi}}_j|) + \sum_{j=1}^{M} \sum_{w=1}^{W} \log(\varphi_{jw}) \tag{2.19}$$

The expression of MML for a finite mixture of EDCM distributions, given a candidate value for M, is then obtained by substituting Eq.(2.19) and Eq.(2.15) into Eq.(2.12).

2.4.3 Algorithm for EDCM Mixture Estimation and Selection

In this section, we summarize the algorithm for estimating the EDCM mixture parameters and selecting the optimal number of consistent components which best describe the data. The input to this algorithm is a dataset $\mathcal{X}$ with N observations each is a W-dimensional count vector representing a document, or an image. Its output is the number of components and estimated parameters. The estimation of the parameters is usually based on the minimization of the MML. However, for estimation, the mixture parameters the Maximum Likelihood approach is very similar to the MML estimates [40]. Note that the initialization of the EDCM mixture parameters can be done using the method of moments equations of DCM given in [75], or with random values.

The complete algorithm for estimation and selection is, thus, summarized is Algorithm 1.

Algorithm 1: Model selection for EDCM mixture model.

Output: Optimal number of components M^*, best model parameters Θ^*

Input: W-dimensional dataset with N vectors $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, and a set of candidates $M = (M_{min}, \ldots, M_{max})$

Initialization of $\Theta^{(0)}$, Set $\tau \leftarrow \tau_{min} (\tau_{min} \ll 1)$.;

for $M_{min} \leq M \leq M_{max}$ **do**

  while $\tau \leq 1$ **do**

    Iterate the two following steps until convergence: ;

    E-step: Compute the posterior probabilities $z_{ij}^{(c)}$ using (2.8) ;

    M-step: Update the mixing weights $\mu_j^{(c)}$ using (2.10);

    Update the EDCM parameter $\boldsymbol{\varphi}_j^{(c)}$ using (2.11);

    Use next temperature parameter $(\tau \leftarrow \tau \times const)$;

  end

  Calculate the associated MML criterion $MessLength(M)$ using (2.12);

end

Select the optimal M^* such that: $M^* = \arg\min_M MessLength(M)$

2.5 Experimental Results

Information explosion is not only creating large amounts of data but also a diverse format of data. For instance, social media platforms offer many possibilities of data formats, including textual data, pictures, videos, sounds, and geolocations. Analyzing different types of data can help in gaining insights into issues, trends, influential actors, and other kinds of information. Several tasks of text classification, for instance, have been studied, including detecting discussed topics in news collection, filtering spam emails, and classifying the sentiment typically in product or movie reviews. Moreover, social media analysis will be affected by the inevitable changes in peoples' tastes, population changes, and many other influences, which makes it important to consider the model capability of coping with the evolution of these data streams. On the other hand, categorizing image databases is very important because it simplifies the task of retrieval by restricting the search for similar images to a smaller domain of the database. Indeed, summarizing the database is very efficient for browsing as knowing the categories of images in a given database allows the user to find the images they are looking for more quickly.

In this section, we demonstrate the effectiveness of the proposed approach via three interesting applications. The first application concerns text classification, the second one involves the detection

of topic novelty, and in the third application, we focus on the problem of image categorization. In our implementations, we have set τ_{min} to 0.04 and $const = 5$, which have been found to be reasonable choices according to [12]. The experiments aim at comparing three count data model-based clustering approaches, namely, the Multinomial mixture (MM), Dirichlet Compound Multinomial (DCM) mixture, to the EDCM. The results that we will present in the following represent the average over 100 runs with different random initialization. Moreover, we compare the results from the MML approach with those obtained for the same EDCM model using other information-theory based techniques.

2.5.1 Text Documents Modeling

Text categorization techniques are essential for classifying new documents and finding interesting information contained within several on-line websites. The applications of text modeling include a number of tasks such as document organization and browsing, corpus summarization, and document classification [76]. The goal of this first application is to investigate the performance of EDCM mixture with MML as a model selection criterion for modeling high dimensional and sparse textual data. The methods that we compare the MML to are: Akaike's Information Criterion (AIC) given by [57]:

$$AIC(M) = -\log(P(\mathcal{X}|\Theta)) + \frac{N_p}{2} \tag{2.20}$$

Minimum Description Length (MDL) given by [58]:

$$MDL(M) = -\log(P(\mathcal{X}|\Theta)) + \frac{1}{2}N_p \log(N) \tag{2.21}$$

Mixture MDL (MMDL) given by [1]:

$$MMDL(M) = -\log(P(\mathcal{X}|\Theta)) + \frac{1}{2}N_p \log(N) + \frac{c}{2}\sum_{j=1}^{M}\log(\mu_j) \tag{2.22}$$

where c is the number of parameters describing each component, equal to $(W + 1)$ in the case of EDCM. The model selected by each method is usually determined according to the candidate number of classes M that yields the minimum value of message length. For our experiments, we

use text datasets that have been considered in the past (see [11, 77]), namely Reuters-10 [1], NIPS [2], WebKB4 and 7Sectors [3].

Reuters-10 is a subset of the well-known corpus Reuters-21578, which is composed of 135 classes with a vocabulary of $15,996$ words. The documents in this dataset are multi labeled, as they may belong to $0, 1$, or many categories. We consider a subset which is composed of the 10 categories having the highest number of class members ($6,775$ and $2,258$ training and testing documents, respectively). Since stop words have already been removed in these collections, we are not removing any additional words. **NIPS** collection contains the OCRed text of all papers published in the 2002 and 2003 NIPS proceedings. This collection has 391 documents in 9 different topics and characterized by $6,871$ words. Papers that are less than 700 words long were eliminated, and stop words were removed. **WebKB4** dataset is a subset of the WebKB dataset containing $4,199$ Web pages gathered from computer science departments of various universities. The considered subset is limited to the four most common categories: Course, Faculty, Project, and Student. The **7sectors** dataset consists of $4,581$ HTML articles in hierarchical order. We considered the initial parent class label to have the documents partitioned into 7 classes. The first step in our preprocessing is removing all stop and rare words from the vocabularies. Then, we perform the feature selection using the Rainbow package [78]. Then, each web page is represented as a vector containing the frequency of occurrence of each word from the term vector. Some statistical properties of the used datasets are summarized in Table (2.1).

Table 2.1: Summarized text datasets properties (N: number of documents, $\overline{n}_d$: average document length, W: vocabulary size, M: true number of classes)

Dataset	N	$\overline{n}_d$	W	M
Reuters-10	$9,033$	193.3	$19,119$	10
NIPS	391	1332.4	$6,871$	9
WebKB4	4199	49.7	$7,786$	4
7sectors	$4,581$	433.2	$4,500$	7

[1] http://kdd.ics.uci.edu/databases/reuters21578
[2] https://cs.nyu.edu/~roweis/data.html
[3] http://www.cs.cmu.edu/~webkb

Table 2.2: Clustering results of text datasets (average $\pm$ slandered error) using a mixture of EDCMs.

Dataset	Model	Accuracy	Precision	Recall	Mutual info.
Reuters-10	EDCM	88.63 ± 0.01	69.36 ± 0.08	80.72 ± 0.07	0.7820 ± 0.04
	DCM	83.44 ± 0.02	74.84 ± 0.05	90.56 ± 0.02	0.7511 ± 0.02
	MM	81.52 ± 0.03	72.31 ± 0.04	82.11 ± 0.02	0.7354 ± 0.06
NIPS	EDCM	90.28 ± 0.03	87.14 ± 0.02	97.70 ± 0.03	0.8406 ± 0.05
	DCM	74.11 ± 0.04	72.58 ± 0.07	83.96 ± 0.03	0.8406 ± 0.07
	MM	69.34 ± 0.17	65.45 ± 0.74	75.31 ± 0.33	79.33 ± 0.16
WebKB4	EDCM	84.31 ± 0.02	84.66 ± 0.06	84.50 ± 0.02	0.7794 ± 0.03
	DCM	82.74 ± 0.06	$83.72 \pm .021$	93.56 ± 0.02	0.7651 ± 0.02
	MM	81.16 ± 0.41	81.20 ± 0.44	82.65 ± 0.50	0.7367 ± 0.35
7sectors	EDCM	91.41 ± 0.05	84.79 ± 0.02	93.75 ± 0.04	0.8321 ± 0.07
	DCM	84.77 ± 0.09	81.26 ± 0.05	89.33 ± 0.04	0.8792 ± 0.06
	MM	75.33 ± 0.02	80.54 ± 0.02	82.78 ± 0.02	0.7877 ± 0.05

The performance of the compared generative models is evaluated using the average classification accuracy, precision and recall averaged at macro level. We have also considered the Mutual Information [12, 65]. Let m_i be the number of documents assigned to class i, and n_j the number of documents with prespecified label j, and c_{ij} the number of documents in class i but misclassified and assigned to class j. With N total documents, define $p_i = m_i/N$, $q_j = n_j/N$, and $r_{ij} = c_{ij}/N$. The MI is then given by:

$$MI = \sum_i \sum_j r_{ij} \log \frac{r_{ij}}{p_i q_j} \tag{2.23}$$

According to Table (2.2), the results achieved using the EDCM for document clustering suggest an outstanding performance based on the four tested datasets. For the Reuters-10 dataset, EDCM outperforms other models with an accuracy of 88.63% compared to 81.52% for MM and 83.44% for DCM. The average accuracy for classifying NIPS using EDCM is 90.28%, which is significantly better than the previously reported results using MM or DCM, given that NIPS document collection has relatively longer documents than the other used datasets. For the WebKB4 collection, the average accuracies achieved are 81.16% and 82.74% (mean plus/minus standard error) by multinomial mixtures and DCM, respectively, and by EDCM is 84.31%. EDCM, once again, outperforms the other tested models for 7sectors datasets with an average accuracy of 91.41%. All differences are statistically significant, as shown by a Student's t-test. Moreover, according to mutual information

reported in Table (2.2), it is clear that EDCM outperforms the other two models (*i.e.*, a Student's t-test shows that the differences in MI between the EDCM, DCM and MM models are statistically significant). The mutual information gained by classifying the Reuters-10 dataset, for example, using EDCM is 0.7820, which is statistically significantly superior to the 0.7511 and 0.7354 by the DCM and MM, respectively.

Fig. (2.1a-2.1d) show the number of clusters found by different selection criteria for Reuters-10, NIPS, WebKB4, and 7sectors datasets, respectively. It is clear that the values of the MML criterion each time agrees with the true number of classes. The number of classes found using MML, MDL, and MMDL with NIPS documents collection is $M = 9$ also agrees with the prespecified number of classes. Moreover, we can see that MML and MMDL found $M = 7$ and $M = 10$, which corresponds to the true number of classes for the 7sectors and Reuters datasets, respectively. For WebKB4, the AIC, MDL, and MMDL criteria failed to find the correct number of clusters, and only MML select $M = 4$ (the true number of classes).

2.5.2 Topic Novelty Detection

2.5.2.1 Online Learning Framework

Novelty detection is an important and challenging task of recognizing that test data differ in some respect from the data that are available during training [37, 79]. It has gained much research attention in application domains involving large datasets acquired from critical systems such as intrusions in electronic security systems of credit card or mobile phone fraud detection, video surveillance, and email and news articles classification. The goal of this application is to apply an online document clustering algorithm based on the EDCM mixture to identify the "novel" dissimilar objects from a sequence of data to previously seen instances. More precisely, the proposed probabilistic model is applied to Topic Detection and Tracking (TDT). The intention is to develop a flexible and accurate online mixture model that correctly identifies the novel topics (detection), and instances of old topics must be correctly classified (tracking) while letting us choose the optimal number of model components simultaneously. For online clustering systems, new coming data is

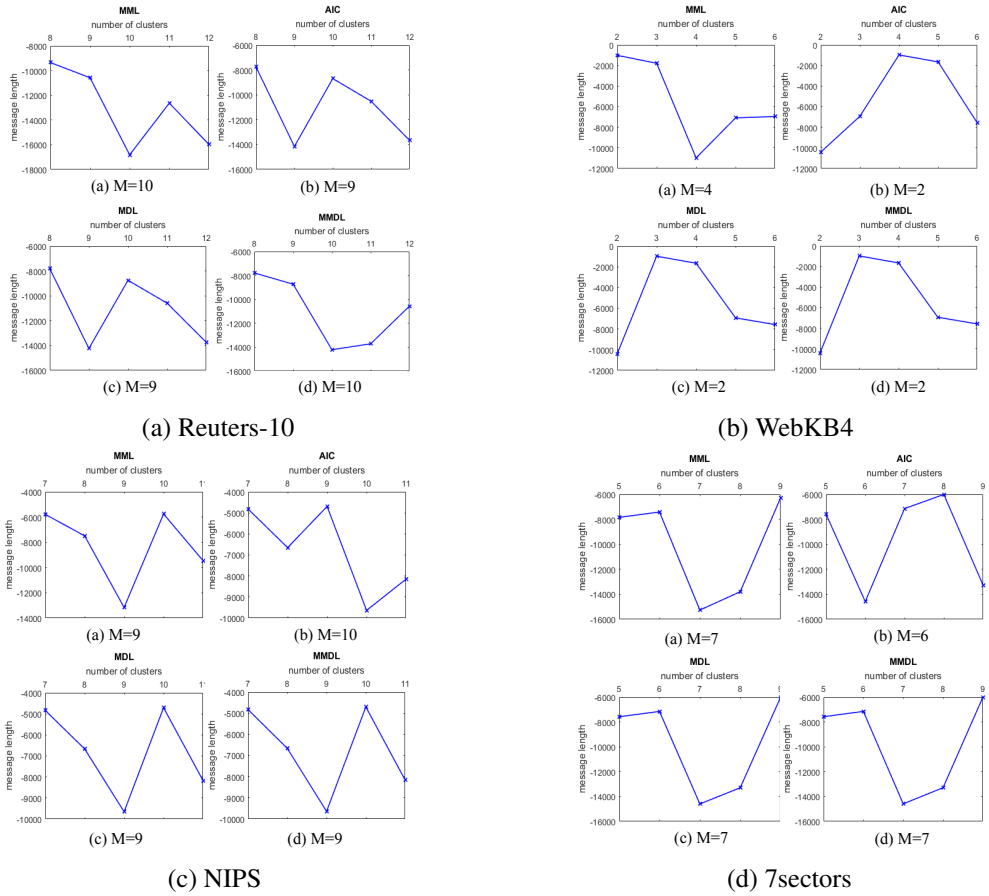


Figure 2.1: Values of the different model selection criteria for the text datasets.

received in online mode, and the model should be updated accordingly without losing its flexibility. Thus, an efficient system should be able to make an important decision regarding whether new classes should be created to represent the new coming data or not.

Formally, assume that at time t we have a dataset $\mathcal{X} = \{\mathbf{X_1}, \ldots, \mathbf{X_N}\}$ of N documents which is represented by an M-component EDCM mixture with parameters $\Theta_M^{(t)}$. Now, at time $t+1$, a new document $\mathbf{X_{N+1}}$ is presented to the model, and the parameters, thus, should be updated incrementally considering the new data which might be assigned to already defined topics or it might be the First Story creating a new cluster. That is, whenever new data is inserted, we find the optimal number of clusters by running the MML model concurrently for models $\{M_{min}, \ldots, M_{max}\}$, and select the candidate value M^* which minimizes the message length. To overcome the problem of slow computation and complexity resulting from the larger range of clusters number candidates,

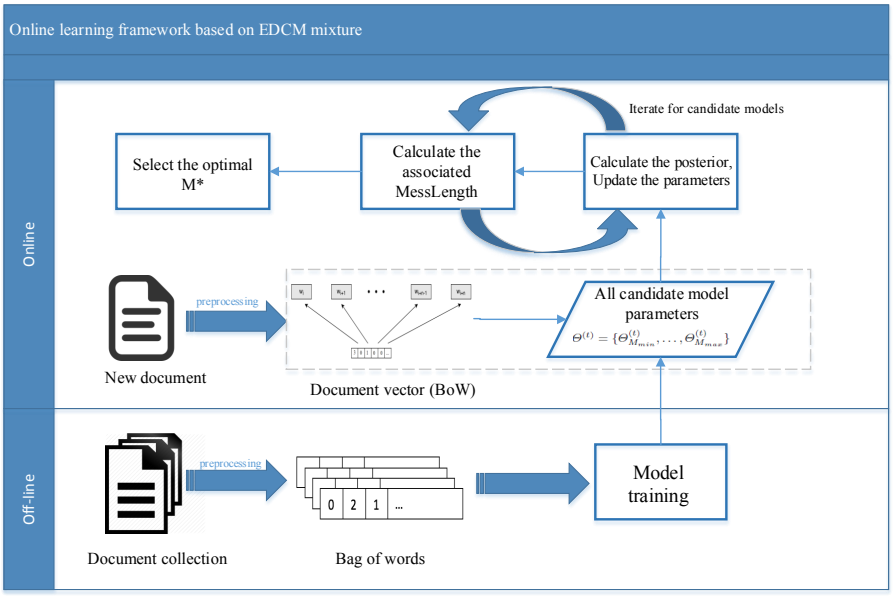


Figure 2.2: Overview of the proposed online framework for topic novelty detection.

we keep all the candidate model fitting with $\{\Theta_{M_{min}}, \ldots, \Theta_{M_{max}}\}$. Fig. (2.2) shows the learning process for both training and updating phases graphically.

For updating the parameters, we use the stochastic ascent gradient parameter updating proposed in [80]. Naturally, we have to keep the constraints $(0 < \mu_j^{(t)} \leq 1)$ and $\sum_{j=1}^{M} \mu_j^{(t)} = 1$. In this regard, new variables $\pi_1, \ldots, \pi_{M-1}$ belongs to $\mathbb{R}$ are considered by introducing the Logit transform,

$$\pi_j = \log \frac{\mu_j}{\mu_M}, \quad j = 1, \ldots, M-1$$

to ensure the unity of the the mixing proportion μ_j. The mixing proportion can be updated as follows:

$$\mu_j^{(t+1)} = \frac{\exp(\pi_j^{(t+1)})}{1 + \sum_{j=1}^{M-1} \exp(\pi_j^{(t+1)})}, j = 1, \ldots, M-1 \tag{2.24}$$

$$\mu_M^{(t+1)} = \frac{1}{1 + \sum_{j=1}^{M-1} \exp(\pi_j^{(t+1)})} \tag{2.25}$$

such that,

$$\pi_j^{(t+1)} = \pi_j^{(t)} + \delta_N(z_{N+1j} - \mu_j^{(t)}),$$

where δ_N is a sequence of positive number that decreases to zero chosen to be $\delta_N = 1/(N+1)$

[36, 80], and z_{N+1j} is the posterior probability of the new coming vector given a set of parameters $\Theta^{(t)}$. Moreover, the model parameters φ_j will be updated according to [80]:

$$\boldsymbol{\varphi}_j^{(t+1)} = \boldsymbol{\varphi}_j^{(t)} + \frac{z_{N+1j}}{N+1} \frac{\partial \log(P(\mathbf{X}_{N+1}, Z_{N+1}|\boldsymbol{\varphi}_j^{(t)}))}{\partial \boldsymbol{\varphi}_j} \tag{2.26}$$

Thus, the complete EDCM mixture updating algorithm for online TDT is as follows:

Algorithm 2: Online learning for EDCM mixture model.

Output: Optimal number of components M^*, updated parameters $\Theta^{*(t+1)}$
Input: At t, W-dimensional dataset with N vectors
$\mathcal{X}^{(t)} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}, \Theta^{(t)} = \{\Theta^{(t)}_{M_{min}}, \ldots, \Theta^{(t)}_{M_{max}}\}$
Input: At $t+1$, new data vector $\mathbf{X}_{N+1}$
while $M_{min} \leq j \leq M_{max}$ **do**
  Compute z_{N+1j} using (2.8), ;
  Assign $\mathbf{X}_{N+1}$ to cluster which maximizes the posterior probability z_{N+1j};
  Update the weights $\mu^{(t+1)}$ using (2.24) and (2.25);
  Update the parameters $\varphi_j^{(t+1)}$ using using (2.26) ;
  Calculate the associated $MessLength$ using (2.12);
end
Select the optimal M^* such that: $M^* = \arg\min_M MessLength(M)$

2.5.2.2 Data, Evaluation Metrics and Results

We illustrate our results on high-dimensional, sparse and challenging real-world datasets, namely, The Topic Detection and Tracking (TDT2) [81], and 20 Newsgroups [4]. In general, the model is built from a training set that is selected to contain no examples, or very few, of the novel class. In our experiments, we initialize the number of components to $M_{max} = 40$, and run the tested algorithms 10 times for evaluation. Moreover, we evaluated the proposed frameworks for topic novelty detection problem using typical evaluation criteria that have been used in the context of text clustering.

TDT-2 dataset contains news stories classified into 96 topics and has been collected in 1998 from six sources: two newswires (Associated Press World Stream and New York Times), two radio programs (Voice of America and Public Radio Internationals The World), and two television programs (CNN and ABC). This corpus is subdivided into three two-month sets: a training set

[4] Both datasets are available at: http://www.cad.zju.edu.cn/home/dengcai/Data/TextData

(Jan-Feb), a development test set (Mar-Apr), and an evaluation set (May-Jun). For preprocessing, documents that belong to several topics have been removed; thus, only 30 topics were left, resulting in $9,394$ with a vocabulary size of $36,771$ words and average document length of 184. This dataset is unbalanced, where some topics have less than 60 documents, while others have over $1,800$ documents. The **20 Newsgroups** contains $18,828$ documents characterized by $61,298$ words with an average document length of 116. The documents in this corpus are fairly distributed over 20 different topics and were collected from UseNet postings over several months in 1993. The datasets characteristics are summarized in Table (2.3).

Table 2.3: Summary of datasets characteristics (N: number of documents, $\overline{n}_d$: average document length, W: vocabulary size, M: true number of classes)

Dataset	N	$\overline{n}_d$	W	M
TDT2	$9,394$	184	$36,771$	30
20newsgroups	$18,828$	116	$61,298$	20

For our experiments, we considered clustering quality and time, which are the natural performance measures for online algorithms [82]. We reported the execution time of the online framework on an Intel(R) Core(TM) i7-6700 Processor PC with the Windows 7 Enterprise Service Pack 1 operating system with a 16 GB main memory. To evaluate the performance, we calculated the F_1 (micro-averaged), $F_1 \in (0, 1)$, measures as follows:

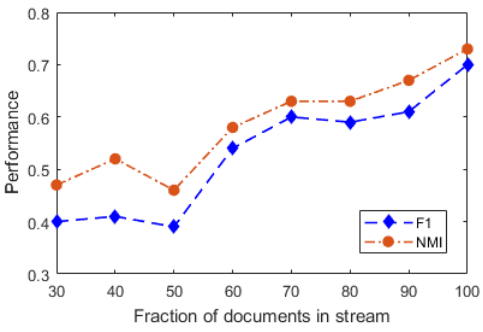
$$F_1(micro - averaged) = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{2.27}$$

In addition, we calculated normalized mutual information (NMI) criterion [82] in order to measure how closely the cluster partitioning could reconstruct the underlying label distribution in the data. For evaluating the performance of topic systems, a standard metric is the cost of detection $cdet$, which combines miss (P_M) and false alarm (P_{FA}) errors into a single number [83].

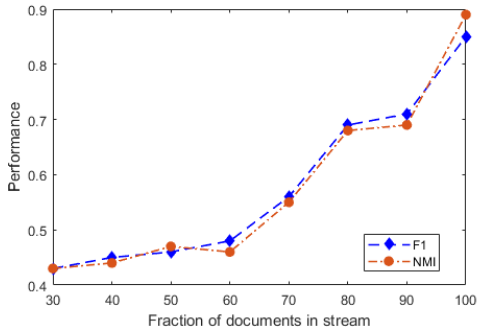
Table (2.4) illustrates the average normalized mutual information (NMI), the average micro F1, the cost of detect and run time results averaged over 10 runs for the two tested datasets. All results are shown in the format of ($average \pm standarderror$). The presented results suggest that

Table 2.4: Performance of different models comparing to online EDCM framework (NMI: normalized mutual information, $F1$: F score micro-averaged, $cdet$: cost of detect).

Dataset	Model	NMI	$F1$	$cdet$	Time
TDT2	EDCM	0.7316 ± 0.04	0.7011 ± 0.05	0.0421 ± 0.01	0.0042
	DCM	0.7044 ± 0.12	0.6805 ± 0.03	0.0533 ± 0.04	0.0112
	MM	0.6797 ± 0.05	0.6650 ± 0.03	0.0687 ± 0.02	0.0183
20newsgroups	EDCM	0.9241 ± 0.06	0.8524 ± 0.04	0.0206 ± 0.06	0.0068
	DCM	0.8534 ± 0.02	0.7518 ± 0.13	0.0490 ± 0.05	0.0480
	MM	0.7449 ± 0.05	0.6587 ± 0.06	0.0563 ± 0.02	0.1280



(a) TDT2 dataset.

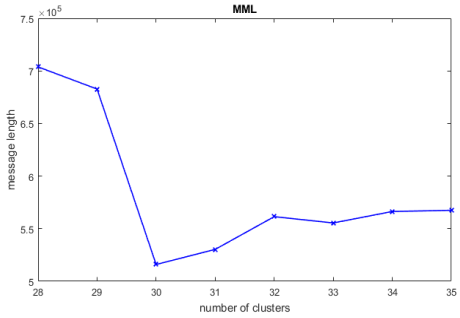


(b) 20Newsgroups dataset.

Figure 2.3: Performance vs. fraction of documents in Online TDT framework based on EDCM mixture model.

the performance of our proposed online TDT framework based on a mixture of EDCM is promising, in terms of time and accuracy, and significantly outperforms both; a mixture of Multinomials and DCMs. The clustering quality measured by the normalized mutual information of 0.7316 and 0.9241 with run time equals to 0.0046 and 0.0068 per story for TDT2 and 20Newsgroups, respectively, offered a faster and more accurate online clustering framework. We compare the performance of the online TDT algorithm on both datasets. First, stories in both datasets have been arranged in chronological order. Then, we selected the oldest 20% articles from each dataset to initialize, where we clustered using the algorithm proposed in [12], and later we used the proposed online algorithm (Algorithm 2), where we insert new story each time until the end. The performance over a different number of stories fed to the system over time is shown in Fig. (2.3). For both datasets, the best $F1$ and NMI are achieved when we insert the whole set of documents.

Moreover, we considered the last class of each dataset as the novel class and compared the



M	Message Length $\mathcal{X}_N$	$\mathcal{X}_{N+1}$
25	826571	843592
26	753717	746300
27	705262	708321
28	693841	703822
29	506807	682755
30	513045	515968
31	538937	530306
32	556147	561438
33	567260	555447
34	580490	566342
35	589004	567472

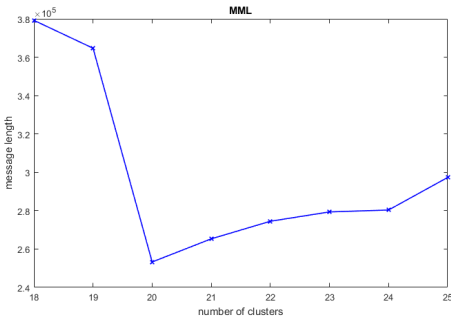
Figure 2.4: Message length values as a function of the number of clusters for the whole data $\mathcal{X}_{N+1}$ in TDT2 dataset.

results of MML for $\mathcal{X}_N$ and whole data after insertion of $\mathcal{X}_{N+1}$. The experiment consists of a training phase and a testing phase. The training phase is the phase during which we build the mixture model that represents the data where we suppose that $\mathcal{X}_N$ come all at time t and build the model according to our algorithm, using the MML to estimate the adequate number of classes. Then, for the testing phase, at time $t+1$ we insert the novel class vectors and update the model according to our proposed online algorithm and use MML again to select the optimal model that represents the whole data $\mathcal{X}_{N+1}$. We can see in Fig. (2.4) and (2.5) that the MML criterion is capable of representing the data, as well as, detect the novel class. The number of classes that minimizes the message length for the training data was $M = 29$ and $M = 19$ for TDT2 and 20Newsgroups datasets, respectively. When the novel vectors inserted, the MML tells that the newly inserted data should be represented by a new component, and the best models selected then were $M = 30$ and $M = 20$ for TDT2 and 20Newsgroups datasets, respectively, which agrees with the exact number of prespecified classes as well.

2.5.3 Hierarchical Image Categorization

2.5.3.1 Hierarchical Clustering Approach

Hierarchical clustering solutions have been primarily obtained using agglomerative algorithms [84, 85], in which objects are initially assigned to their own cluster, and then pairs of clusters are repeatedly merged until the whole tree is formed. We are proposing a hierarchical clustering algorithm based on the probabilistic distance between the EDCM mixture components. Probabilistic distance



M	Message Length $\mathcal{X}_N$	$\mathcal{X}_{N+1}$
15	410186	422942
16	351677	411357
17	345552	388388
18	274877	379207
19	211488	364685
20	218465	253263
21	219130	265324
22	220022	274482
23	221065	279352
24	227574	280352
25	257980	297323

Figure 2.5: Message length values as a function of the number of clusters for the whole data $\mathcal{X}_{N+1}$ in 20Newsgroups dataset.

measures between two probability distributions are significant metrics to evaluate the similarity for data of statistical nature. If the parameters of two Probability Density Functions (PDF) are known or can reliably be estimated; a quantitative value can be calculated to assess how far or close the two distributions are from each other [86]. The Kullback Leibler (KL) Divergence [87, 88], also known as the relative entropy, is a widely used approach in statistics to measure the similarity between two density distributions. The KL divergence between two EDCM distributions with parameters θ_{j1} and θ_{j2}, with $n = \sum_{i=1}^{N} x_i$, and $s_j = \sum_{w=1}^{W} \varphi_{jw}$, is given by (see Appendix 2):

$$KL\Big(P(X|\theta_{j1}), P(X|\theta_{j2})\Big) = \log\Big[\frac{\Gamma(s_{j1})\Gamma(s_{j2}+n)}{\Gamma(s_{j1}+n)\Gamma(s_{j2})}\Big] + \sum_{w=1}^{W}\Big(\log(\varphi_{j1}) - \log(\varphi_{j2})\Big)\Big(\Psi(s_{j1}+n) - \Psi(s_{j1})\Big) \tag{2.28}$$

The input to the hierarchical algorithm is a $M \times M$ similarity matrix, where M is the optimal number of classes. The approach involves finding the least dissimilar pair of clusters in the current clustering, the shortest distance, and merge them into a single cluster to form the next clustering level with $M - 1$ classes. Updating the dissimilarity and merging the closest pair of clusters are repetitively done until all objects are in one cluster. We clarify the generation of the dendrogram by means of the graphical representation in Fig. (2.6).

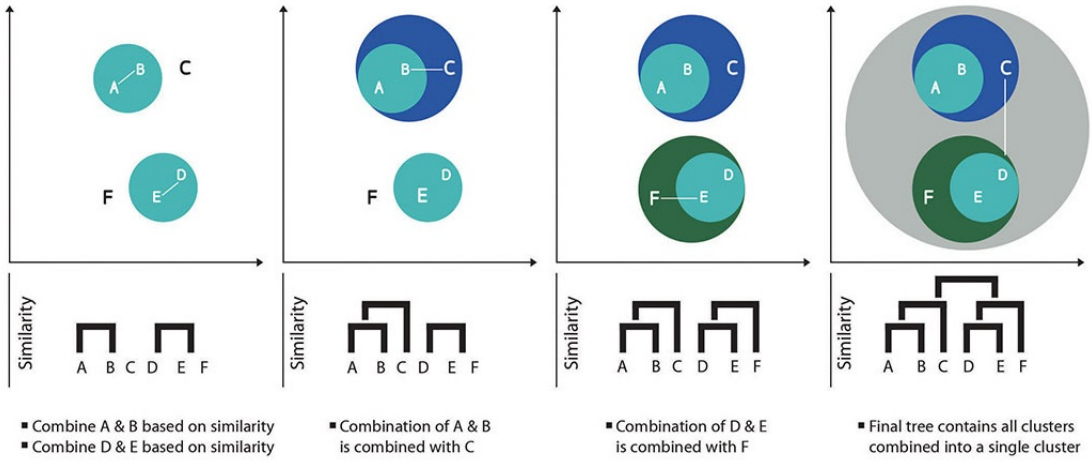


Figure 2.6: The graphical representation of the hierarchical clustering resulting in a tree-structured graph(dendrogram).

2.5.3.2 Data Representation and Results

Recent successful approaches in image classification and retrieval are inspired by the text retrieval systems. That is, the local image patches are considered as the visual equivalents of individual words and images are, thus, represented as a "bag of visual words" or "bag of features" [89]. Our baseline system builds upon the bag-of-features approach, which has demonstrated comparable or better results than other approaches for object-based image classification. First, a set of local image patches "Keypoints" are extracted, and a vector of visual descriptors is evaluated on each patch independently using the Scale-Invariant Feature Transform (SIFT) [90]. The resulting distribution of descriptors is then quantized into a number of homogeneous clusters using an unsupervised clustering approach, typically a k-means algorithm [91], where the centroid of each cluster is treated as a visual word. The representation is then obtained by assigning the descriptors (for features extracted from a novel image) to the closest visual word (Euclidean distance) resulting in a histogram of frequencies that can, then, be used for the categorization.

We evaluate our model performance on three different image datasets: CIFAR-10 [92], Natural Scene [93], and PPMI [94]. The **CIFAR-10** dataset collected by researchers at MIT and NYU over the span of six months. The dataset consists of $60,000$ natural-color images of size 32×32 collected using several search engines, including Google, Flickr, and Altavista, based on $79,000$ search terms. The images belonging to 10 completely mutually exclusive categories are split into $50,000$ for

Figure 2.7: Example images of the PPMI dataset.

training images and $10,000$ test images ($1,000$ images per class). **Natural Scene** Contains 13 categories available in grayscale only. The average size of each image is 250×250 pixels. The dataset includes $3,859$ images, classified as 13 categories: 360 coasts, 328 forest, 374 mountain, 410 open country, 260 highway, 308 inside of cities, 356 tall buildings, 292 streets, 241 suburb residence, 174 bedroom, 151 kitchen, 289 living room, and 216 office. Each category of scenes was split randomly into two separate sets of images, $80 : 20$ for training (learning the codebook) and testing (creating the BOF representation). **PPMI**, People-playing-musical-instruments, consists of 7 different musical instruments bassoon, erhu, flute, French horn, guitar, saxophone, and violin (Fig. 2.7). Each class includes $\sim$ 300 highly diverse and cluttered images of humans playing or holding the instruments. The dataset is already split randomly into two separate sets of images, half for training and a half for testing.

For constructing the codebook and obtaining the bag-of-features representation, we learn $25k$, $22k$, and $15k$ visual vocabulary from CIFAR-10, Natural Scene, and PPMI datasets, respectively. For each dataset, we compare the different model selection criteria for the same model parameters (Fig. 2.8). We observe that all the methods worked well in the case of the CIFAR-10 dataset. All four criteria selected $M = 10$, which is the true number of classes. However, for the Natural Scene dataset, only MML and MMDL selected the true number of classes $M = 13$, where AIC and MDL selected $M = 12$ and $M = 11$, respectively. For PPMI, the true number of clusters $M = 7$ was again selected by MML and MMDL. These results make it clear that the MML model selection criterion outperforms other methods indicated by the true number of classes always selected by

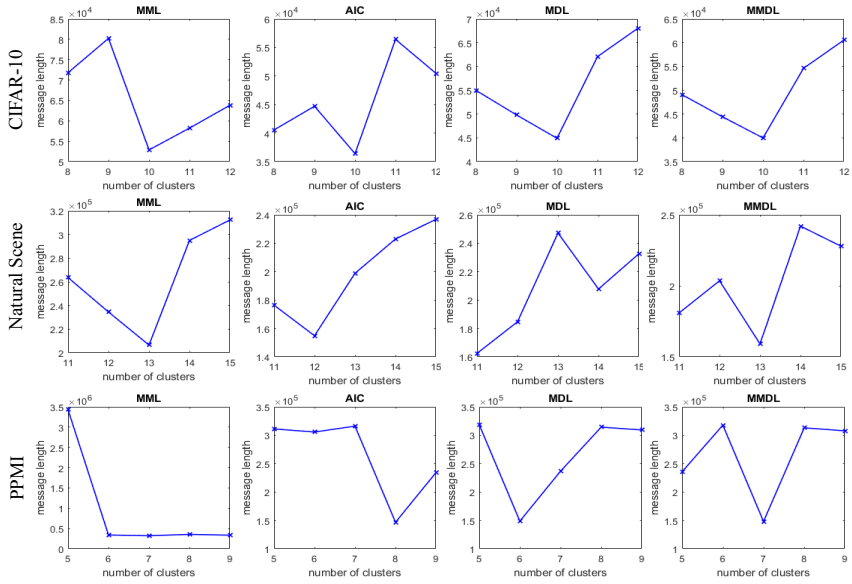


Figure 2.8: Number of clusters found by the different criteria for the different image datasets.

Table 2.5: Comparison of image databases categorization results using different clustering approaches.

Dataset	M	W	MM	DCM	EDCM	H-EDCM
CIFAR-10	10	$25k$	89.05 ± 0.08	91.46 ± 0.12	94.40 ± 0.36	98.77 ± 0.02
Natural Scene	13	$22k$	81.24 ± 0.03	85.34 ± 0.13	93.45 ± 0.41	99.51 ± 0.03
PPMI	7	$15k$	83.79 ± 0.04	88.50 ± 0.02	97.11 ± 0.16	99.64 ± 0.05

MML with a mixture of EDCM distributions.

The overall accuracy for image categorization obtained using the different generative models are shown in Table (2.5). Clearly, EDCM outperforms both MM and DCM for the three tested images datasets. According to the reported results, the average categorization accuracies for CIFAR-10 are 94.40% using EDCM, 91.46%, and 89.05% for DCM and MM, respectively. Moreover, the overall accuracy of categorizing the Natural Scene dataset using EDCM mixture is 93.45%, which is significantly better than 85.34% by DCM and 81.24% by MM mixture. Similarly, EDCM achieves significantly superior performance in categorizing the PPMI dataset with an overall accuracy of 97.11%, compared to 83.79% and 88.50% by MM and DCM, respectively.

Furthermore, we evaluate the proposed hierarchical clustering algorithm by conducting another experiment where we compare both the traditional and hierarchical clustering approaches on the

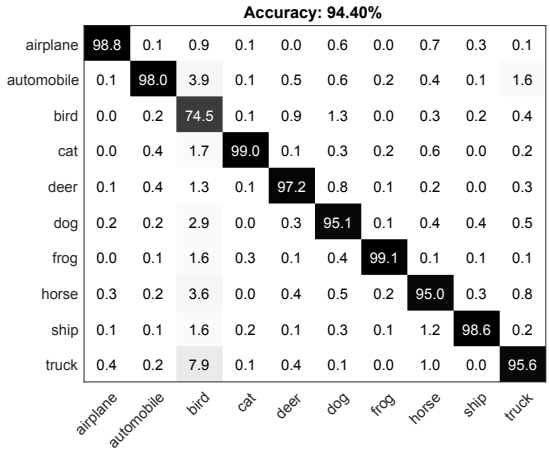


Figure 2.9: Confusion matrix for clustering the CIFAR-10 dataset using an EDCM mixture model.

three different datasets. We first categorize the images into the optimal number of classes M, and we can further establish some relationship among the categories by looking at KL divergence among them and merge those most similar categories repetitively. Table (2.5) shows the correct number of classes M, the vocabulary size W and the accuracy for clustering in M classes using the different generative models, as well as, the accuracy after all classes are merged (H-EDCM) using the proposed hierarchical algorithm for the different datasets.

Moreover, we measure the intra-class performance of the EDCM mixture for categorizing the images in the 10 classes of CIFAR-10 using the confusion matrix (Fig. 2.9). Each entry (i, j) of the confusion matrix denotes the percentage of images in class i that are assigned into class j. From this figure, we can see that the average categorization accuracy was 94.40% (an error rate of 5.60%) for this database. The best classified objects are *frog* and *cat* with a performance of 99.1 percent and 99.0 percent, respectively. Then, we evaluated the similarity between each pair of distributions. When the KL divergence between them is the minimum, the categories are close to each other and can be merged into one class to form the tree (Fig. 2.10). The evaluation of pairwise similarity was repetitively done, and after all the categories are hierarchically related, the accuracy improved to 98.77%. A Student's t-test shows that the improvement is statistically significant (p-values between 0.012 and 0.044).

Fig. 2.11(a) shows example images from the Natural Scene dataset. The most difficult scenes to classify are *tall building*. There is confusion between the *street* and *forest* scenes and between

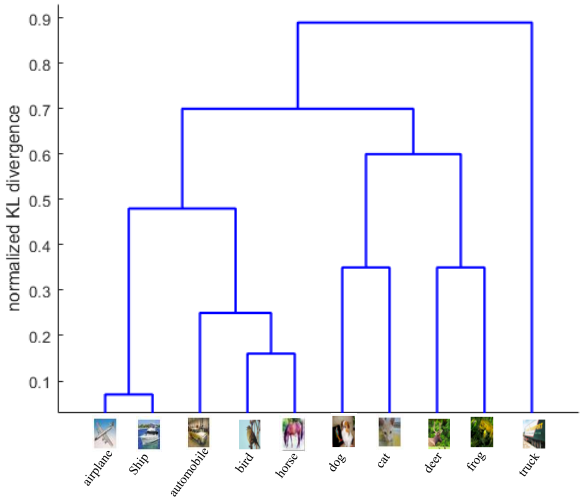


Figure 2.10: Dendrogram showing the closest categories in CIFAR-10 based on KL divergence.

the *living room* and *office* scenes which were also the most similar categories. We established the relationship among the similar categories and evaluated the accuracy of $M-1$ classes after each pair of clusters are merged into a larger one (Fig. 2.11(b)). Clearly, the EDCM mixture performance is very much improved by the hierarchical clustering approach (accuracy increased from 93.45% to 99.51%). The difference in accuracy is statistically significant, as shown by Student's t-test (p-values between 0.032 and 0.039).

The PPMI clustering accuracy achieved by the EDCM mixture was 97.11% (an error rate of 2.89%). From the confusion matrix (Fig. 2.12(a)), we can see that most difficult instruments to classify are *bassoon* and *erhu* with a performance of 84.4 percent and 98.7 percent, respectively. Considering the KL divergence between each pair of distributions in the PPMI dataset, we could establish the hierarchical relationships as when the distributions are most similar; the categories are also close to each other on the dendrogram (Fig. 2.12(b)). For example, the closest categories are *flute* and *erhu*, then *saxophone* and *French horn*. As shown in Table (2.5), the accuracy of categorizing PPMI using the EDCM mixture has been improved from 97.11% to 99.64% using the hierarchical clustering approach. This difference is, once again, statistically significant according to the Student's t-test (p-values between 0.030 and 0.042).

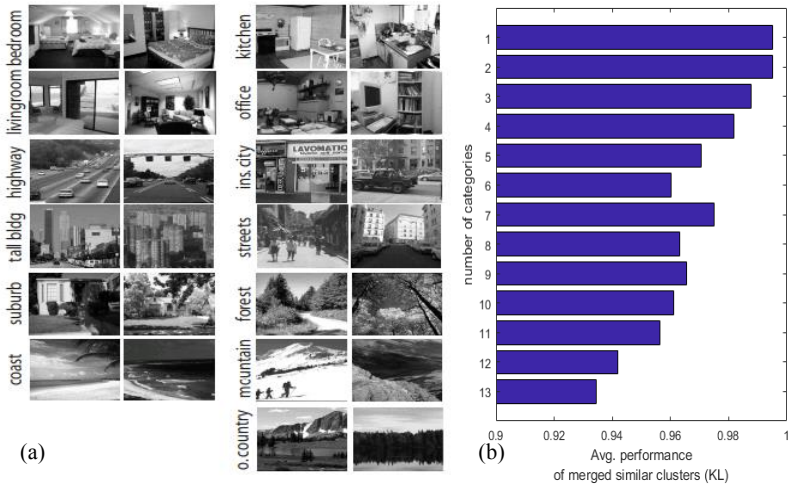


Figure 2.11: (a) Example images of the 13 categories Natural Scene dataset (b) Average accuracy of each level of the tree after merging similar clusters based on KL divergence.

2.6 Conclusion

In this work, we have proposed an MML-based approach to select the model that best represents the data based on a finite EDCM mixture. The deterministic-annealing expectation-maximization algorithm has been used to estimate the parameters of this model, and the obtained results, when applied on real data, show its merit as an unsupervised learning model for clustering count data. The proposed model-selection approach based on EDCM was applied to three challenging real-world applications, including text mining, topic detection, and tracking, and hierarchical image categorization. For each application, a comprehensive performance evaluation of the model and selection criterion was given using different large and widely-used datasets. We believe that the mixture of EDCM distributions with the proposed MML approach offers strong modeling capabilities for many other applications that involve high dimensional and sparse count data.

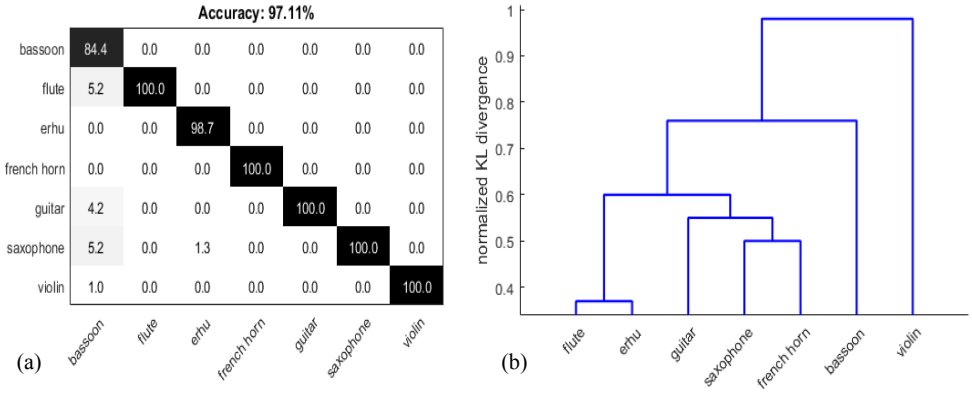


Figure 2.12: (a) Confusion matrix for the PPMI (b) Dendrogram showing the closest categories based on KL divergence.

Appendix 1: Proof of Eq.(2.15)- The Fisher Information Matrix for EDCM

We have the negative of log-likelihood function as:

$$
\begin{aligned}
-\mathcal{L}(\mathcal{X}_j|\boldsymbol{\varphi}_j) &= -\log\left(\prod_{d=l}^{l+\eta_j-1}\mathcal{EDCM}(\mathbf{X}_d|\boldsymbol{\varphi}_j)\right) \\
&= \eta_j(-\log\Gamma(s_j)) + \sum_{d=l}^{l+\eta_j-1}\log\Gamma(s_j+n_d) - \sum_{w:x_{dw}\geq 1}\log\varphi_{jw} + \log x_{dw} \qquad (2.29)
\end{aligned}
$$

Then, the first order derivative of the negative log-likelihood, also called the Fisher score function:

$$
-\frac{\partial\mathcal{L}(\mathcal{X}_j|\varphi_j)}{\partial\varphi_{jw}} = \eta_j(-\Psi(s_j)) + \sum_{d=l}^{l+\eta_j-1}\Psi(s_j+n_d) - \sum_{d=l}^{l+\eta_j-1} I(x_{dw}\geq 1)\frac{1}{\varphi_{jw}} \qquad (2.30)
$$

where Ψ is the digamma function. Then,

$$
\begin{aligned}
-\frac{\partial^2\mathcal{L}(\mathcal{X}_j|\varphi_j)}{\partial\varphi_{jw}^2} &= \eta_j(-\Psi'(s_j)) + \sum_{d=l}^{l+\eta_j-1}\Psi'(s_j+n_d) \\
&\quad + \sum_{d=l}^{l+\eta_j-1} I(x_{dw}\geq 1)\frac{1}{\varphi_{jw}^2} \qquad (2.31)
\end{aligned}
$$

and:

$$-\frac{\partial^2 \mathcal{L}(\mathcal{X}_j|\varphi_j)}{\partial\varphi_{jw1}\partial\varphi_{jw2}} = \eta_j(-\Psi'(s_j)) + \sum_{d=l}^{l+\eta_j-1} \Psi'(s_j+n_d), w_1 \neq w_2 \tag{2.32}$$

where Ψ' is the trigamma function. We remark that $F(\boldsymbol{\varphi}_j)$ can be written as:

$$F(\boldsymbol{\varphi}_j) = D_j + \gamma_j \mathbf{A}\mathbf{A}^T \tag{2.33}$$

where $D = diag\Big[\sum_{d=l}^{l+\eta_j-1} I(x_{dw} \geq 1)\frac{1}{\varphi_{j1}^2}), \ldots, \sum_{d=l}^{l+\eta_j-1} I(x_{dw} \geq 1)\frac{1}{\varphi_{jW}^2})\Big]$, $\gamma = \eta_j(-\Psi'(s_j)) + \sum_{d=l}^{l+\eta_j-1} \Psi'(s_j+n_d)$, and $A^T = 1$. Then, according to(Theorem 8.4.3) given by Graybill [95], the determinant of the Fisher information matrix $F(\varphi_j)$ is:

$$|F(\boldsymbol{\varphi}_j)| = \left(1 + \gamma_j \sum_{w=1}^{W} \frac{a_{jw}^2}{D_{jw}}\right) \prod_{w=1}^{W} D_{jw} \tag{2.34}$$

By substituting Eq.(2.34) and Eq.(2.14) into Eq.(2.13), we obtain:

$$|F(\Theta)| \simeq \frac{N}{\prod_{j=1}^{M} \mu_j} \prod_{j=1}^{M} \left[\left(1 + \gamma_j \sum_{w=1}^{W} \frac{a_{jw}^2}{D_{jw}}\right) \prod_{w=1}^{W} D_{jw}\right] \tag{2.35}$$

Then, taking the log gives Eq.(2.15).

Appendix 2: Proof of Eq.(2.28)- KL Divergence Between Two EDCM Distributions

The KL divergence between two distributions that belong to the exponential family is defined as [67]:

$$KL\Big(P(X|\theta_{j1}), P(X|\theta_{j2})\Big) = \Phi(\theta_{j1}) - \Phi(\theta_{j2}) + \big(G(\theta_{j1}) - G(\theta_{j2})\big) E_{\theta_{j1}}[T(X)] \tag{2.36}$$

where E_θ is the expectation with respect to $P(X|\theta)$. We have:

$$\Phi(\theta) = \frac{\Gamma(s)}{\Gamma(s+n)} \tag{2.37}$$

$$G(\theta) = \log(\varphi_{jw}) \tag{2.38}$$

$$T(X) = I(x_w \geq 1) \tag{2.39}$$

Moreover, we have the following [67]:

$$E_\theta[T(X)] = -\Phi'(\theta) \tag{2.40}$$

Thus, according to Eqs.(2.37 and 2.39), we have:

$$E_\theta\Big[I(x_w \geq 1)\Big] = -\frac{\partial\Phi(\theta)}{\partial\varphi_{jw}} = \Psi(s+n) - \Psi(s) \tag{2.41}$$

substituting Eqs. (2.41, 2.37 and 2.38) in Eq.(2.36) gives Eq.(2.28).

Chapter 3

A Novel Scaled Dirichlet-based Statistical Framework for Count Data Modeling: Unsupervised Learning and Exponential Approximation

The multinomial distribution and the Dirichlet Compound Multinomial (DCM) are widely accepted to model count data. However, recent research showed that the Dirichlet is not the best choice as prior to multinomial. Thus, we propose a novel model called the Multinomial Scaled Dirichlet (MSD) distribution that is the composition of the scaled Dirichlet distribution and the multinomial. Moreover, to improve the computation efficiency in high-dimensional spaces, we propose to approximate the MSD as a member of the exponential family. The performance evaluation of the proposed models is conducted through a set of extensive empirical experiments on challenging applications, namely, text classification, facial expression recognition, and texture images clustering. The results show that the proposed model, and its approximation, strive to achieve higher accuracy compared to the state-of-the-art generative models for count data clustering, while the approximation EMSD is many times faster than the corresponding MSD.

3.1 Introduction

Count data appear in many domains in machine learning and computer vision applications. Consider, for instance, textual collections, or image modeling and clustering where each document or image can be represented by a vector of frequencies of words or visual words, respectively. Real texts systematically exhibit the burstiness phenomena; if a word appears once in a document, it is much more likely to appear again [7, 54]. This phenomenon is not limited to text and can also be observed in images with visual words [96].

Modeling the probabilities of words occurrences improves classification performance and information retrieval accuracy. Multinomial distributions fail to capture this phenomenon well, as was shown in [9]. Hierarchical Bayesian modeling was proposed as an appropriate and efficient solution to address this issue by introducing the Dirichlet distribution as a prior to the Multinomial, which results in the Dirichlet Compound Multinomial (DCM) [9]. The hierarchical approach of DCM considers the count vector to be generated by a multinomial distribution whose parameters are generated by the Dirichlet distribution. That is, in a specific document, for example, the Multinomial is linked to particular sub-topics, and thus, it makes the emission of some words more likely than others. This gives it the ability to handle burstiness, even for rare words. This composition is based mainly on the fact that the Dirichlet is a conjugate to the multinomial offers numerous computational advantages [55]. The Dirichlet, however, has some drawbacks, including its very restrictive negative covariance structure, inconsiderate relations between categories, and its poor parameterization [63, 97, 98]. Thus, different efficient alternatives to the DCM have been lately proposed; namely, the Multinomial Generalized Dirichlet Distribution (MGD) [10], and the Multinomial Beta-Liouville Distribution (MBL) [11]. This paper is based on another alternative model called the Multinomial Scaled Dirichlet (MSD) distribution, which we have previously proposed in [23]. The new proposed model MSD is the composition of the scaled Dirichlet distribution and the multinomial in the same way that DCM, MGD, MBL are the compositions of the Dirichlet, the generalized Dirichlet, and the Beta-Liouville, respectively, with the multinomial. For clustering, we considered a finite mixture model that permits a formal approach for unsupervised learning.

On the other hand, in bag-of-words, or bag-of-visual-words, representation, many features occur

only once, and many more do not occur at all, as each observation contains only a small subset of the vocabulary. This is referred to as the sparsity nature resulting in many of the entries being zero. Thus, text documents and images are usually represented as high-dimensional and sparse vectors, *i.e.,* a few thousand dimensions with a sparsity of 95 to 99% [81]. The sparseness of data is heavily studied in the literature, where many techniques have been proposed to optimize data representation for more efficient and accurate clustering (see, for example,[99]). Among the successful approaches, Elkan [12] has shown that the estimation algorithm of the exponential-family approximation to the DCM, EDCM, is much faster than the corresponding algorithm with DCM and has the ability to model the burstiness phenomenon well even for rare words. Indeed, exponential families of distributions offer several appealing statistical and computational properties [14, 65]. For instance, sufficiency retains the essential information in a data set regarding the parameters which reduce the computation time, especially for sparse high-dimensional data. The new proposed model MSD shares similar problems to the ones with DCM, including that it does not belong to the exponential family, its expression lacks intuitiveness, and its parameters cannot be estimated quickly. Thus, we derive a new distribution that is a close approximation to the MSD. The proposed approximation is a member of the exponential family of distributions that we called (EMSD). Moreover, for determining the number of components in the EMSD mixture, we develop a Minimum Message Length (MML) criterion as an efficient unsupervised learning algorithm for clustering high-dimensional and sparse count data. The consideration of the MML approach is inspired by its impressive performance for model selection in the case of EDCM [22]. By means of some challenging applications, we show that both MSD and EMSD are better suited than the multinomial and DCM for modeling count data.

The rest of the paper is organized as follows. In Section 3.2, we review DCM and other alternatives proposed in the literature. Section 3.3, introduces a new prior to the multinomial, namely, the scaled Dirichlet, and applies the expectation-maximization (EM) algorithm with the MSD mixture. Next, Section 3.4 discusses the approximation in detail where we derive a new family of distributions that we called EMSD and the deterministic annealing EM (DAEM) approach to learn the EMSD parameters. The MML expression for the EMSD mixture is detailed in Section 3.5. Finally, Section 3.6 is devoted to experimental results, and Section 3.7 concludes the paper.

3.2 Hierarchical Bayesian Models for Count Data

3.2.1 Dirichlet Compound Mutinomial (DCM)

Multinomial distribution, the multivariate generalization of the Binomial distribution, is widely used in count data clustering. For instance, in modeling text documents using Multinomial distribution, an individual document is represented as a vector of word counts (bag-of-words representation). The assumptions for the Multinomial model are the length of document n (in tokens) is known, and the occurrences of the words are independent of each other.

Define $\mathbf{X} = (x_1, \ldots, x_W)$ as a sparse vector of counts representing a document, or an image, where x_w is the frequency of a word, or visual word, w. Then, the probability of $\mathbf{X}$ that it follows a Multinomial distribution with parameters $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_W)$, is given by:

$$\mathcal{M}(\mathbf{X}|\boldsymbol{\rho}) = \frac{n!}{\prod_{w=1}^{W} x_w!} \prod_{w=1}^{W} \rho_w^{x_w} \tag{3.1}$$

where W is the vocabulary size, and $n = \sum_{w=1}^{W} x_w$.

The Dirichlet distribution, with a set of parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_W)$, is defined as:

$$\mathcal{D}(\boldsymbol{\rho}|\boldsymbol{\alpha}) = \frac{\Gamma(a)}{\prod_{w=1}^{W} \Gamma(\alpha_w)} \prod_{w=1}^{W} \rho_w^{\alpha_w - 1} \tag{3.2}$$

where $a = \sum_{w=1}^{W} \alpha_w$. Then, the DCM is the marginal distribution given by the following integration [9]:

$$\begin{aligned} \mathcal{DCM}(\mathbf{X}|\boldsymbol{\alpha}) &= \int_{\rho} \mathcal{M}(\mathbf{X}|\boldsymbol{\rho}) \mathcal{D}(\boldsymbol{\rho}|\boldsymbol{\alpha}) d\rho \\ &= \frac{n!}{\prod_{w=1}^{W} (x_w)!} \frac{\Gamma(a)}{\Gamma(\sum_{w=1}^{W} x_w + \alpha_w)} \prod_{w=1}^{W} \frac{\Gamma(x_w + \alpha_w)}{\Gamma(\alpha_w)} \end{aligned} \tag{3.3}$$

We can note that compared to the multinomial, the DCM has one extra degree of freedom, since its parameters are not constrained to sum up to one, which makes it more practical [61, 63].

3.2.2 Efficient Alternative Priors

Although the Dirichlet distribution is a natural conjugate prior to the multinomial likelihood, and it exhibits many convenient mathematical properties, it is not the most appropriate solution. Hence, other distributions have been proposed in the literature to be used as a prior for the multinomial. Dirichlet distribution, for instance, has a very restrictive negative covariance structure, and the variables with the same mean must have the same variance [98, 100]. These properties make its use as a prior in the case of positively correlated data inappropriate. Recent works show that the generalized Dirichlet has many convenient properties that make it more useful and practical, as prior to the multinomial than the Dirichlet in real-life applications [10, 100]. Bouguila [11] later introduced another alternative based on Liouville family distributions, which is in contrast with the Dirichlet and, like the generalized Dirichlet, can have positive or negative covariance. In addition, like the Dirichlet and the generalized Dirichlet, the Liouville distribution of the second kind is a conjugate to the Multinomial distribution [100]. Other interesting properties of the Liouville distribution are discussed in [101, 102]. In this work, we look at other limitations of the Dirichlet distribution. For instance, Dirichlet does not take into account relative positions between categories or multinomial cells [98]. Moreover, it has an inadequate parameterization that limits its ability to better model variance and covariance [97]. Thus, we are proposing the choice of a more flexible prior to the Multinomial that can help to resolve these issues, which is a generalization of the Dirichlet called scaled Dirichlet distribution [98, 103].

3.3 The Proposed Model

In this section, we discuss in details the proposed model that we called Multinomial Scaled Dirichlet (MSD) distribution, based on introducing a new prior to the multinomial, namely, the scaled Dirichlet.

3.3.1 Multinomial Scaled Dirichlet (MSD)

In this work, we look at some limitations of the Dirichlet distribution. For instance, Dirichlet does not take into account relative positions between categories or Multinomial cells [98]. Moreover, it has a poor parameterization that limits its ability to better model variance and covariance [97]. Thus, we are proposing the choice of a more flexible prior to the Multinomial that can help to resolve these issues, which is a generalization of the Dirichlet called Scaled Dirichlet distribution [98, 103]. The Scaled Dirichlet is a generalization of the Dirichlet distribution, which is the distribution of a random vector obtained after applying the perturbation and powering operations to a Dirichlet random composition. These operations define a vector-space structure in the simplex and play the same role as sum and product by scalars in real space [103].

The scaled Dirichlet is a generalization of the Dirichlet distribution obtained after applying the perturbation and powering operations to a Dirichlet random composition. These operations define a vector-space structure in the simplex and play the same role as sum and product by scalars in real space [103]. In dimension W, the scaled Dirichlet with a set of parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_W)$ which is the shape parameter, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_W)$ which is the scale parameter, is defined by [104, 105]:

$$\mathcal{SD}(\boldsymbol{\rho}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\Gamma(a)}{\prod_{w=1}^{W} \Gamma(\alpha_w)} \frac{\prod_{w=1}^{W} \beta_w^{\alpha_w} \rho_w^{\alpha_w - 1}}{\left(\sum_{w=1}^{W} \beta_w \rho_w \right)^a} \tag{3.4}$$

where Γ denotes the Gamma function, and $a = \sum_{w=1}^{W} \alpha_w$.

The shape parameter $\boldsymbol{\alpha}$ simply describes the form or shape of the scaled Dirichlet distribution, and its flexibility is very significant in finding patterns and shapes inherent in a dataset. The scale parameter $\boldsymbol{\beta}$ controls how the density plot is spread out where the shape of the density is invariant, irrespective of the value of a constant or uniform scale parameter. Note that the Dirichlet distribution is a special case of the scaled Dirichlet that can be obtained when all elements of the vector $\boldsymbol{\beta}$ are equal to a common constant [103]. Thus, the scaled Dirichlet includes the Dirichlet as a special case. Compared to the Dirichlet, the scaled Dirichlet, has W extra parameters, which enhances the model flexibility [106, 107]. The proper parameterization of scaled Dirichlet gives it the ability to

better model variance and covariance [97]. Moreover, unlike Dirichlet, the scaled Dirichlet takes into account relative positions between categories or multinomial cells [98, 103]. These properties make the scaled Dirichlet a more flexible choice of a prior to Multinomial.

The composition of the Multinomial and scaled Dirichlet is, thus, obtained by integrating over ρ, which gives the marginal distribution of $\mathbf{X}$, as follows:

$$\begin{aligned} \mathcal{MSD}(\mathbf{X}|\boldsymbol{\alpha},\boldsymbol{\beta}) &= \int_{\rho} \mathcal{M}(\mathbf{X}|\boldsymbol{\rho})\mathcal{SD}(\boldsymbol{\rho}|\boldsymbol{\theta})d\rho \\ &= \frac{n!}{\prod_{w=1}^{W} x_w!} \frac{\Gamma(a)}{\Gamma(a+n)} \frac{}{\prod_{w=1}^{W} \beta_w^{x_w}} \prod_{w=1}^{W} \frac{\Gamma(\alpha_w + x_w)}{\Gamma(\alpha_w)} \end{aligned} \tag{3.5}$$

This equation is obtained by using the fact that $\int_{\rho} \mathcal{SD}(\boldsymbol{\rho}|\boldsymbol{\alpha},\boldsymbol{\beta}) = 1$, and applying the following empirically tested approximation: $\left(\sum_{w=1}^{W} \beta_w \, \rho_w\right)^{\sum_{w=1}^{W} \alpha_w} \simeq \prod_{w=1}^{W} \beta_w^{\alpha_w}$, given a common constant value for $\boldsymbol{\beta}$ (see Appendix 1). Moreover, by setting $\beta_1 = \beta_2 = \cdots = \beta_W = 1$, Eq.(3.5) is reduced to the DCM.

3.3.2 The Multinomial Scaled Dirichlet Mixture Estimation

Statistical-based approaches are powerful and widely used in generative learning processes to abstract the complexity of a huge amount of information. One major approach based on statistics is the finite mixture models that are used to model data sampled from a finite number of homogeneous subpopulations, where the whole model is formed by a weighted sum of the subgroups' densities. Given an observed dataset $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ with N data instances, each is a W-dimensional vector $\mathbf{X}_i = (x_{i1}, \ldots, x_{iW})$ drawn from a superposition of M multinomial scaled Dirichlet densities of the form:

$$P(\mathbf{X}_i|\boldsymbol{\pi},\boldsymbol{\theta}) = \sum_{j=1}^{M} \pi_j \, \mathcal{MSD}(\mathbf{X}_i|\boldsymbol{\theta}_j) \tag{3.6}$$

where π_j ($0 < \pi_j < 1$ and $\sum_{j=1}^{M} \pi_j = 1$) are the mixing proportions. Each $\mathcal{MSD}(\mathbf{X}_i|\boldsymbol{\theta}_j)$ is called a component of the mixture, and has its own parameters $\boldsymbol{\theta}_j = \{\boldsymbol{\alpha_j}, \boldsymbol{\beta_j}\}$, where $\boldsymbol{\alpha}_j = (\alpha_{j1}, \ldots, \alpha_{jW})$, and $\boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jW})$.

Next, we introduce a M-dimensional binary random vector $\mathbf{Z_i} = (z_{i1}, \ldots, z_{iM})$ to each data

vector $\mathbf{X_i}$, where $\sum_{j=1}^{M} z_{ij} = 1$, and z_{ij} is a latent variable works as an indicator equal to one if $\mathbf{X}_i$ belongs to cluster j and zero, otherwise. The complete data at this case are $(\mathcal{X}, \mathcal{Z}|\Theta)$, where $\mathcal{X}$ represents a set of observed variables, the complete set of all latent variables and parameters is denoted by $\Theta = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_M, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_M, \pi_1, \ldots, \pi_M)$. Thus, the complete data log-likelihood corresponding to a M-component mixture is given by:

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{M} z_{ij} \Big(\log P(\mathbf{X}_i|\theta_j) + \log \pi_j \Big). \tag{3.7}$$

For learning a mixture model, the Expectation Maximization (EM) algorithm is the most popular approach which generates a sequence of models with non-decreasing log-likelihood on the data. In EM, estimation is broken down into a two-step iteration (E-step and M-step) using the notion of incomplete data which produces a sequence of estimates $\{\Theta^{(t)}, t = 0, 1, 2, \ldots\}$. The posterior probabilities will be computed in the **E-step**, as:

$$\hat{z}_{ij}^{(t)} = \frac{\pi_j^{(t)} \; P(\mathbf{X}_i|\theta_j^{(t)})}{\sum\limits_{j=1}^{M} \pi_j^{(t)} \; P(\mathbf{X}_i|\theta_j^{(t)})} \tag{3.8}$$

In the **M-step**, the parameters estimates will be updated according to:

$$\Theta^{(t+1)} = \arg\max_{\Theta} \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta, \Theta^{(t)}) \tag{3.9}$$

when maximizing (3.9), we obtain:

$$\pi_j^{(t+1)} = \frac{1}{N} \sum_{i=1}^{N} \hat{z}_{ij}^{(t)} \tag{3.10}$$

The maximum likelihood parameter estimate is obtained by taking the derivative of the log-likelihood function and find Θ when the derivative is equal to zero. However, we do not obtain a closed-form solution for the $\boldsymbol{\alpha}_j$ and $\boldsymbol{\beta}_j$ parameters. We, therefore, use the Newton-Raphson method

expressed as:

$$\theta_j^{new} = \theta_j^{old} - H^{-1}G, \tag{3.11}$$

where H is the Hessian matrix associated with $\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)$, and G is the gradient vector associated with the first-order derivatives (see Appendix 2), where the complete block Hessian matrix H_j has to be transformed to its inverse before it can be used in the Newton-Raphson maximization.

Indeed, the EM algorithm highly depends on the initialization to handle the local maxima problem resulting from the multimodal nature of the likelihood function when we deal with mixture models [108]. Thus, proper initialization is needed in order to achieve optimal performance. Thus, we initialize the π_j parameter using the K-means algorithm, and to initialize the model parameters, we make use of the method of moments. In the case of the multinomial scaled Dirichlet distribution, a closed-form solution for its moment equations does not exist. Thus, we initialize the $\boldsymbol{\alpha}_j$ vector using the moments' equations of the DCM distribution [75], while the $\boldsymbol{\beta}_j$ vector is initialized with equal scaling (a vector of ones). Parameters will be then updated during the EM iterations to take their natural values in relation to the observed data. Then, the complete algorithm for learning the MSD mixture parameters is summarized in Algorithm 3.

Algorithm 3: MSD mixture model parameters estimation.

Output: Model parameters $\{\theta_j\}_{j=1}^{M}$
Input: Dataset $\mathcal{X} = \{X_1, \ldots, X_N\}$, each is a W-dimensional vector, a pre-specified number of clusters M

```
Initialize the mixing weights π_j using K-means ;
Initialize the shape parameters α_j using method of moments ;
Initialize the scale parameter vector β_j with a vector of ones;
while Convergence criteria is not reached do
    for i = 1 to N do
        for j = 1 to M do
            Compute the posterior probabilities p(j|X_i, θ_j) using equation (3.8) ;
        end
    end
    for j = 1 to M do
        Update the mixing proportion π_j using Eq.(3.10);
        Update the parameters θ_j = {α_j, β_j} using Eq.(3.11);
    end
end
```

3.4 Approximating the MSD

In this section we derive a new distribution that is an approximation to the MSD. We call the new distribution EMSD and it is, unlike the MSD, a member of the exponential family.

3.4.1 An Exponential-family Approximation to MSD (EMSD)

Given the sparsity nature of datasets represented using bag-of-words, or bag-of-visual-words, it should be possible to evaluate the probability as a function of non-zero x_w values only for computational efficiency. That is, the value of $x_w! = 1$, $\beta_w^{x_w} = 1$ and $\Gamma(\alpha_w + x_w)/\Gamma(\alpha_w) = 1$ when $x_w = 0$. The MSD distribution in this case is given by:

$$\mathcal{MSD}(\mathbf{X}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{n!}{\prod\limits_{w:x_w \geq 1} x_w!} \frac{\Gamma(a)}{\Gamma(a+n)} \frac{}{\prod\limits_{w:x_w \geq 1} \beta_w^{x_w}} \prod_{w:x_w \geq 1} \frac{\Gamma(\alpha_w + x_w)}{\Gamma(\alpha_w)} \tag{3.12}$$

In case of high dimensional data when the parameters are really small, we can use the approximation given in [12] to replace $\Gamma(\alpha_w + x_w)/\Gamma(\alpha_w)$ by $\Gamma(x_w)\alpha_w$ in the previous equation. Using the fact that if x is an integer then $x! = x(x-1)!$, we can further simplify to obtain the new distribution that we call (EMSD):

$$\mathcal{EMSD}(\mathbf{X}) = \frac{n!}{\prod\limits_{w:x_w \geq 1} x_w} \frac{\Gamma(s)}{\Gamma(s+n)} \prod_{w:x_w \geq 1} \frac{\lambda_w}{\nu_w^{x_w}} \tag{3.13}$$

where $s = \sum_{w=1}^{W} \lambda_w$. We denote the EMSD parameters as λ_w instead of α_w, and ν_w instead of β_w to distinguish them from the MSD parameters for clarity.

This form shows that EMSD allows multiple appearances of the same word to have higher probability. Moreover, it distinguishes between word types and word tokens, as modeling both frequencies is beneficial for capturing the statistical properties of natural languages [66]. Similar to DCM and EDCM, the maximum likelihood estimates of MSD and EMSD are sensitive to which words appear in which documents, while Multinomial ignores the type-token distinction (*i.e.* the Multinomial parameters are the same regardless documents boundaries in the collection). We can

rewrite Eq.(3.13) in exponential family form as:

$$\mathcal{EMSD}(\mathbf{X}) = \Big(\prod_{w:x_w \geq 1} x_w \Big)^{-1} n! \frac{\Gamma(s)}{\Gamma(s+n)} \times \exp\Big[\sum_{w=1}^{W} I(x_w \geq 1) \log(\lambda_w) - x_w \log(\nu_w) \Big] \tag{3.14}$$

where $I(x_w \geq 1)$ is an indicator that represents whether the word w appears at least once in the frequency vector $\mathbf{X}$.

3.4.2 Mixture of EMSDs Learning

For learning a mixture of EMSD distribution, we propose to use the deterministic annealing EM (DAEM) [53], which has been efficiently used to avoid the major issues of initialization dependency and poor local maxima in regular EM. The deterministic annealing approach uses multiple phases, each with a value of temperature parameters set, where the final Θ parameters in each phase are used as initial values in the next one. The annealing process begins at a high temperature where the function is smoothed and has only one global optimum point. As the temperature decreases, the function shape gradually approaches the original objective, so the DAEM continually tracks the new optimum point until it finds the best one. Moreover, exploring a larger region of parameter space through the slow EM convergence is an important factor in the good performance of soft clustering algorithms [110]. Practically, slower convergence makes the weights z_{ij} further away from zero and one, thus they reflect the membership uncertainty more realistically [12].

When applying the deterministic annealing procedure, posterior probabilities will be computed in the E-step, as:

$$z_{ij}^{(t)} = \frac{\Big(\pi_j^{(t)}\ P(\mathbf{X}_i|\theta_j^{(t)})\Big)^{\tau}}{\sum_{j=1}^{M} \Big(\pi_j^{(t)}\ P(\mathbf{X}_i|\theta_j^{(t)})\Big)^{\tau}} \tag{3.15}$$

where $\tau = \frac{1}{T}$, and T corresponds to the computational temperature [1]. Then, the maximum likelihood parameter estimates can be obtained by taking the derivative of the log-likelihood function

[1] Experimentally, we have concluded that using a three-phases annealing, with $T \in [1, 5, 25]$, is a reasonable choice that gives good results, which is the same confirmation reached in [10, 12].

associated with the complete data $\mathcal{Q}(\mathcal{X}, \mathcal{Z}|\Theta)$ and find Θ when the derivative is equal to zero. Thus, setting partial derivative of the log-likelihood with respect to $\lambda_{jw}, w = 1, \ldots, W$, to zero and solving for λ_{jw}, gives:

$$\lambda_{jw} = \frac{\sum_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}{\sum_{i=1}^{N} z_{ij} \Psi(s_j + n_i) - K\Psi(s_j)} \tag{3.16}$$

where $K = \sum_{i=1}^{N} z_{ij}$. Then, we can compute s_j by summing each side of Equation (3.16) over all words, giving:

$$s_j = \frac{\sum_{w=1}^{W} \sum_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}{\sum_{i=1}^{N} z_{ij} \Psi(s_j + n_i) - K\Psi(s_j)} \tag{3.17}$$

The numerator in this case is the number of times a word w appears at least once in any vector of the dataset. Note that this equation can be solved numerically efficiently by Newton's method as it involves only a single unknown, s_j. Having s_j in hand, Eq.(3.16) can be used directly to compute each individual λ_{jw} [2]. Furthermore, by setting the partial derivative of the log-likelihood with respect to $\nu_{jw}, w = 1, \ldots, W$, to zero and solving for ν_{jw}, we obtain:

$$\nu_{jw} = -\sum_{i=1}^{N} z_{ij} I(x_{iw} \geq 1) \tag{3.18}$$

3.5 MML Criterion for EMSD Mixture

Minimum Message Length (MML) is a selection criterion based on evaluating statistical models according to their ability to compress a message containing the data. High compression is obtained by forming good models of data to be coded [70]. For each model in the model space, the message includes two parts. The first part encodes the model using only prior information about the model and no information about the data. The second part encodes only the data in a way that makes use

[2] Eqs. (3.16) and (3.17) are similar to (5) and (6) in [12] for estimating the parameters of the EDCM.

of the model encoded in the first part [69].

Let $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ a set of data controlled by a mixture of EMSD distributions with parameters Θ. According to information theory, the optimal number of clusters M is the candidate value, which minimizes the amount of information, measured in $nats$ using the natural logarithm, to transmit $\mathcal{X}$ efficiently from a sender to a receiver [71]. The formula for the message length for a mixture of distributions, with N_p free parameters, is given by [40, 41]:

$$\begin{aligned} MessLength \simeq & -\log(h(\Theta)) - \log(P(\mathcal{X}|\Theta)) \\ & + \frac{1}{2}\log(|F(\Theta)|) + \frac{N_p}{2}\Big(1 + \log(k_{N_p})\Big) \end{aligned} \tag{3.19}$$

where $h(\Theta)$ is the prior probability, $P(\mathcal{X}|\Theta)$ is the likelihood for the complete data set, $|F(\Theta)|$ is the determinant of the expected Fisher information matrix, and k_{N_p} is the optimal quantization lattice constant for $\mathbb{R}^{N_p}$ [72]. When $N_p = 1$ the value of $k_1 = 1/12 \simeq 0.083$, and as N_p grows, k_{N_p} tends to the asymptotic value given by $\dfrac{1}{2\pi e} \simeq 0.05855$, which can be approximated by $\frac{1}{12}$ [41].

The complete-data Fisher information matrix in mixture models has a block-diagonal structure, and its determinant is given as the product of the determinant of the Fisher information of $\boldsymbol{\theta}_j$ for each component and the determinant of the Fisher information of mixing parameters vector $\boldsymbol{\pi}$ [1, 40]. Given that the mixing proportions satisfy the requirement $\sum_{j=1}^{M} \pi_j = 1$, it is possible to consider its Fisher information as a series of trails; each has M possible outcomes. Thus, the number of trails of the jth cluster is a multinomial distribution with parameters $(\pi_1, \ldots, \pi_M)$ [40, 59]. The $|F(\theta_j)|$ is the determinant of the Fisher information matrix of the expected second partial derivatives of the negative log-likelihood [41]. As proposed in [1], the Fisher information matrix in case of a mixture model can be computed after the data vectors have been assigned to their respective clusters. Let $\mathcal{X}_j = \{\mathbf{X}_l, \ldots, \mathbf{X}_{l+\eta_j-1}\}$ be the data elements in the jth cluster where $l \leq N$ and η_j the number of the documents generated by the jth topic with parameters $\boldsymbol{\theta}_j$. The negative of the log-likelihood function given the vector $\boldsymbol{\theta}_j = \{\lambda_j, \nu_j\}$ and $s_j = \sum_{w=1}^{W} \lambda_{jw}$ of a single EMSD distribution is $-\mathcal{Q}(\mathcal{X}_j|\boldsymbol{\theta}_j)$. The complete Fisher matrix for each component $F(\theta_j)$ has a block structure (see Appendix 3); therefore, we compute the determinant of each block matrix using the solution provided in [111]. Thus, we can show that $\log(|F(\Theta)|)$ in case of finite EMSD mixture

model is given by:

$$\log(|F(\Theta)|) = \log(N) - \sum_{j=1}^{M} \log(\pi_j) + \sum_{j=1}^{M} \log(|F(\theta_j)|) \tag{3.20}$$

The capability of the MML criterion is controlled by the choice of prior distribution $h(\Theta)$ for the parameters of the EMSD. In case of mixture models,we make a general assumption in mixture models that the parameters of the different components as a prior are independent from the mixing probabilities, and the components of $h(\boldsymbol{\lambda}_j)$ and $h(\boldsymbol{\nu}_j)$ are independent as well [73], that is:

$$h(\Theta) = h(\boldsymbol{\pi}) \prod_{j=1}^{M} \prod_{w=1}^{W} h(\lambda_{jw}) h(\nu_{jw}) \tag{3.21}$$

Knowing that the vector $\boldsymbol{\pi}$ is defined on the simplex $\{(\pi_1, \ldots, \pi_M) : \sum_{j=1}^{M} \pi_j = 1\}$, then the Dirichlet distribution is a natural choice as a prior for the mixing probabilities. The choice of a constant Dirichlet parameters (a vector of ones) gives a uniform prior as follows [40, 41]:

$$h(\boldsymbol{\mu}) = \Gamma(M) = (M-1)! \tag{3.22}$$

For calculating $h(\boldsymbol{\lambda})$, $h(\boldsymbol{\nu})$ and in the absence of other knowledge about the $\lambda_{jw}, \nu_{jw}, w = 1, \ldots, W$, we assume that $h(\lambda_{jw})$ and $h(\nu_{jw})$ are locally uniform over the ranges $[0, e^6 \frac{|\hat{\lambda_j}|}{\lambda_{jw}}]$ and $[0, e^6 \frac{|\hat{\nu_j}|}{\nu_{jw}}]$ where $\hat{\lambda_j}$ and $\hat{\nu_j}$ are the estimated vector. We choose to use a simple uniform prior, which is known to give good results, in accordance with Ockham's razor [74], as:

$$h(\lambda_{jw}) = \frac{e^{-6}\lambda_{jw}}{|\hat{\boldsymbol{\lambda}}_j|}, h(\nu_{jw}) = \frac{e^{-6}\nu_{jw}}{|\hat{\boldsymbol{\nu}}_j|} \tag{3.23}$$

Thus, substituting Eq.(3.23) and Eq.(3.22) into Eq.(3.21), and taking the log we obtain:

$$\begin{aligned} \log(h(\Theta)) = &\sum_{j=1}^{M} \log(j) - 12MW - W \sum_{j=1}^{M} \log(|\hat{\boldsymbol{\lambda}}_j|) \\ &+ \sum_{j=1}^{M} \sum_{w=1}^{W} \log(\lambda_{jw}) - W \sum_{j=1}^{M} \log(|\hat{\boldsymbol{\nu}}_j|) + \sum_{j=1}^{M} \sum_{w=1}^{W} \log(\nu_{jw}) \end{aligned} \tag{3.24}$$

The expression of MML for a finite mixture of EMSD distributions, given a candidate value for M, is then obtained by substituting Eq.(3.24) and Eq.(3.20) into Eq.(3.19).

3.6 Experimental Results

In this section, we demonstrate the effectiveness of the proposed approaches via four interesting real-world applications. The first application concerns text classification, the second one involves the recognition of facial expression, in the third application we focus on the problem of categorizing natural scene images, and the last application concerns the clustering of texture images. The experiments aim at comparing the Multinomial mixture (MM), the DCM, the MGD, and the MBL to our proposed models MSD and EMSD. The results that we report in the following represent the average over 20 runs of the different learning algorithms.

3.6.1 Text Classification

Text classification is the task of automatically assigning predefined categories to documents written in natural languages. Several tasks of text classification have been studied, each of which deals with different types of documents and categories, such as detecting discussed topics in news collection, filtering spam emails, and classifying the sentiment typically in product or movie reviews. The performance of the proposed approaches is compared using precision and recall averaged at the macro level, and F score averaged at macro and micro levels [112], to evaluate how the models perform overall across the tested sets of data. We have also considered the mutual information (MI) [12, 110], to quantify how much the assigned classes by an algorithm agrees with the pre-specified ones. We use datasets that have been widely considered in the past (see, for instance, [113, 114]), and/or used in the previously with the tested models to ensure fair comparison. The text datasets used in our experiments are IMDB[3], Reuters-21578 [4], WebKB4, and 20Newsgroups[5].

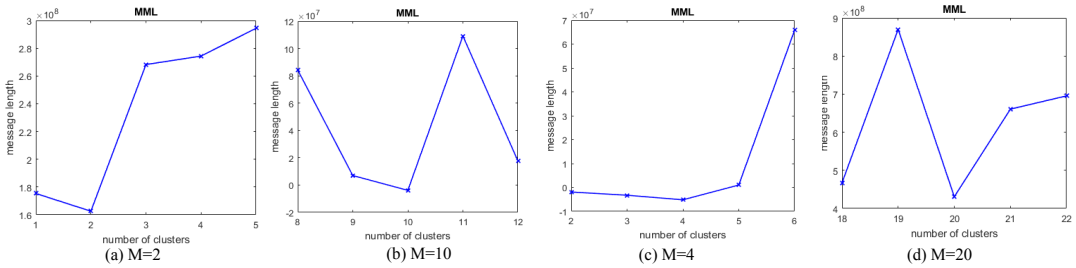
IMDB (movie reviews) contains positive and negative sentiments [115]. Ratings on IMDB are given as star values $\in \{1, 2, \ldots, 10\}$, which were linearly mapped to $[0, 1]$ to use as document

[3] http://ai.stanford.edu/~amaas/data/sentiment/
[4] http://www.daviddlewis.com/resources/testcollections/reuters21578/
[5] http://www.cs.cmu.edu/~webkb/

Table 3.1: Description of the text data sets and comparison of the running time for MSD and EMSD (N:number of documents, W: vocabulary size, $\overline{n_i}$: average document length, M: number of classes).

Dataset	N	W	$\overline{n_i}$	M	T_{MSD}	T_{EMSD}
IMDB	$50,000$	$76,340$	115.25	2	926.44	350.25
Reuters-10	$9,033$	$19,119$	193.9	10	495.6	84.3
WebKB4	$4,199$	$7,786$	502.2	4	135.4	15.2
20News.	$18,846$	$61,298$	136.7	20	876.2	136.5

labels; negative and positive, respectively. We used a union of the training and testing sets having around $25,000$ samples from each positive/negative group with $76,340$ unique words in total. The second data set used is a subset of the well-known corpus **Reuters-21578**, which is composed of 135 classes with a vocabulary of $15,996$ words. For our experiments, we consider a subset which is composed of the 10 categories having the highest number of class members ($6,775$ and $2,258$ training and testing documents are considered for this subset, respectively). **WebKB4** data set is a subset of the WebKB data set containing $4,199$ Web pages gathered from computer science departments of various universities. The considered subset is limited to the four most common categories: Course, Faculty, Project, and Student. Finally, the **20 Newsgroups** which contains $18,828$ documents characterized by $61,298$ words, and the documents are fairly distributed over 20 different news topics. Note that the experiments are done directly here (*i.e.,* we do not separate the data set into training and testing sets). The Rainbow package [78] was used to read the text files and perform the feature selection considering words with the highest average mutual information after removing all stop and rare words (less than 50 occurrences in our experiments). For sentiment analysis, certain stop words (e.g., negating words) are indicative, so traditional stop word removal was not used in the IMDB dataset. Each text file is then represented as a vector containing the occurrence frequency for each word from the vocabulary. Some statistical properties of the used data sets are summarized in Table (3.1), where we also show the running time (in seconds) for each data set using MSD and EMSD. The reported execution time is for optimized MATLAB2017a codes run on an Intel(R) Core(TM) i7-6700 Processor PC has the Windows 7 Enterprise Service Pack 1 operating system with a 16 GB main memory. Fig. (3.1) shows the number of clusters selected by our EMSD algorithm, and we can observe that the exact number of clusters ($M = 2, M =$



(a) M=2 (b) M=10 (c) M=4 (d) M=20

Figure 3.1: Number of clusters found by the MML criterion for the text data sets (a)IMBD, (b)Reuters-10, (c)WebKB4, and (d)20newsgroups.

Table 3.2: Classification results for IMDB dataset using MSD/EMSD mixture models.

Measures	MM	DCM	MGD	MBL	**MSD**	**EMSD**
Precision	0.74	0.71	0.75	0.76	0.84	0.85
Recall	0.82	0.89	0.81	0.81	0.84	0.86
Macro F	0.75	0.84	0.82	0.82	0.84	0.86
Micro F	0.76	0.84	0.82	0.82	0.84	0.85
Mutual Info.	0.73	0.88	0.89	0.80	0.88	0.89

$10, M = 4$, and $M = 20$) was selected for the IMBD, Reuters-10, WebKB4 and 20newsgroups data set, respectively.

Table 3.3: Classification results for Reuters-10 dataset using MSD/EMSD mixture models.

Measures	MM	DCM	MGD	MBL	**MSD**	**EMSD**
Precision	0.72	0.74	0.77	0.76	0.76	0.80
Recall	0.92	0.93	0.95	0.94	0.91	0.94
Macro F	0.81	0.83	0.86	0.86	0.83	0.87
Micro F	0.88	0.90	0.93	0.94	0.90	0.94
Mutual Info.	0.89	0.88	0.75	0.79	0.88	0.94

The classification results for the four data sets are given in Table (3.2-3.5), reported as the average percentage of performance metrics over the 20 runs. According to the F measures and Mutual information in this table, the new proposed model MSD behaves similarly to MGD and MBL (*i.e.*, a Student's t-test shows that the difference in performance is not statistically significant: p-values are 0.1380 and 0.1422 for F measure and MI, respectively). However, MSD is shown to outperform Multinomial, and DCM as the differences between the MSD these models are statistically significant (*i.e.*, p-values between 0.0004 and 0.0071 for the different runs). On the other hand, the

exponential family approximation to MSD (EMSD) is outperforming all the other tested models as shown by a Student's t-test (*i.e.*, p-values are 0.0275 and 0.0012 for the difference between MSD and EMSD according to F measure and MI, respectively). Moreover, compared to MSD, the EMSD based classification is between 3 and 9 times faster for the different data text sets.

Table 3.4: Classification results for WebKB4 dataset using MSD/EMSD mixture models.

Measures	MM	DCM	MGD	MBL	**MSD**	**EMSD**
Precision	0.81	0.84	0.86	0.89	0.76	0.90
Recall	0.82	0.84	0.87	0.88	0.91	0.89
Macro F	0.82	0.89	0.89	0.89	0.83	0.89
Micro F	0.82	0.88	0.88	0.88	0.90	0.89
Mutual Info.	0.73	0.77	0.84	0.89	0.89	0.91

Table 3.5: Classification results for 20newsgroups dataset using MSD/EMSD mixture models.

Measures	MM	DCM	MGD	MBL	**MSD**	**EMSD**
Precision	0.85	0.86	0.87	0.86	0.88	0.95
Recall	0.86	0.88	0.90	0.91	0.84	0.96
Macro F	0.86	0.87	0.88	0.88	0.86	0.96
Micro F	0.85	0.86	0.87	0.88	0.89	0.95
Mutual Info.	0.87	0.88	0.88	0.86	0.86	0.97

3.6.2 Facial Expression Recognition

Facial expressions are the facial changes in response to a person's internal emotional states, intentions, or social communications. The automatic analysis and recognition of facial motions and facial features changes from visual information have been the topic of extensive research. Automatic facial expression analysis can be applied in many areas such as emotion and paralinguistic communication, clinical psychology, psychiatry, neurology, pain assessment, lie detection, intelligent environments, and human-computer interface (HCI) [116]. An efficient computer facial expression analysis system needs to analyze the facial actions regardless of context, culture, gender, and so on. Generally, six basic facial expressions have been considered, namely anger, disgust, fear, happiness, sadness, and surprise. In this work, the performance of our approaches was evaluated on a challenging facial expression datasets, namely, MMI [117], CK+ [118].

Figure 3.2: Sample facial expression images from the MMI dataset.

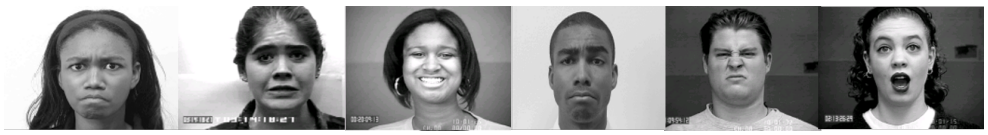


Figure 3.3: Sample facial expression images from the CK+ dataset.

The **MMI** dataset includes 19 different faces of students and research staff members of both genders (44% female), ranging in age from 19 to 62, having either a European, Asian, or South American ethnic background. Currently, it contains $2,894$ image sequences where each image sequence has a neutral face at the beginning and the end, and each with the size of 720×576 pixels. We selected the sequences that could be labeled as one of the six basic emotions. Removing the natural faces resulting in $1,140$ images (150 Anger, 212 Disgust, 150 Fear, 255 Happiness, 192 Sadness, and 181 Surprise). Sample images from the MMI dataset with different facial expressions are shown in Fig. (3.2). The **Extended Cohn-Kanade (CK+)** dataset consists of facial behavior of 210 adults 18 to 50 years of age, 69% female, 81% Euro-American, 13% Afro-American, and 6% other groups. Image sequences were digitized into either 640×490 or 640×480 pixel arrays with 8-bit gray-scale value. We included all posed expressions that could be labeled as one of the 6 basic emotion categories, which is about $4,000$ images (342 Anger, 503 Disgust, 417 Fear, 993 Happiness, 893 Sadness, and 852 Surprise). Fig. (3.3) shows examples of the emotion images from the CK+ dataset. For each dataset, we randomly divided the selected image sequences into two partitions: one for constructing the visual vocabulary, the other for representation.

The recognition accuracy of the facial expression, obtained by applying the different approaches to the considered datasets, is shown in Table (3.6). The classification accuracy for detecting facial

Table 3.6: Facial expression recognition results (average %) using MSD/EMSD mixture models.

Dataset	MM	DCM	MGD	MBL	**MSD**	**EMSD**
MMI	67.74	74.25	80.00	80.65	80.00	99.42
CK+	70.94	77.41	77.84	73.09	77.84	97.74

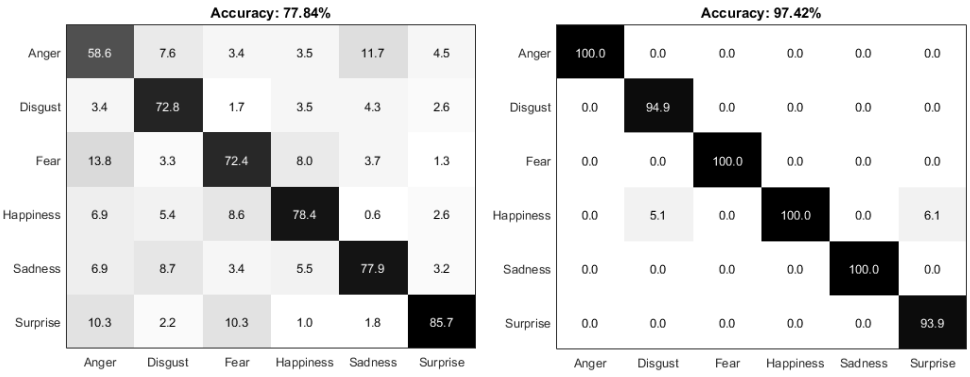
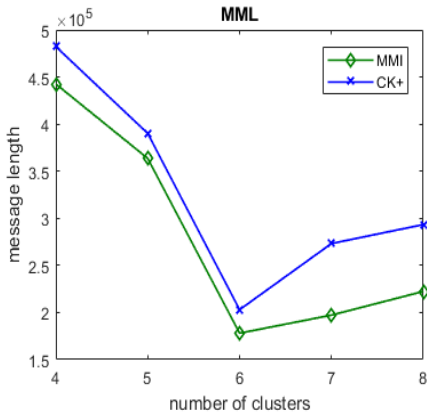


Figure 3.4: Confusion matrix for the six categories in the CK+ dataset using MSD (left), and EMSD (right).

expression in MMI and CK+ datasets using MSD are (80.00%) and (77.84%), respectively. According to the results, it is clear that the difference between the MGD, MBL, and MSD is again not statistically significant (p-values between 0.1823 and 0.3211 for the different runs) while these three models outperform Multinomial for the MMI and outperforming both Multinomial and DCM for CK+ dataset. Moreover, EMSD clearly performs better on both datasets. Figure (3.4) below shows the confusion matrix obtained by the proposed MSD and EMSD for the CK+ database.

From Fig. (3.4), we can see that the average categorization accuracy using MSD is (77.84%), an error rate of (22.16%), for this database. The best classified expressions are *surprise* and *happiness* with a performance of 85.7 percent and 78.4 percent, respectively. Using EMSD, the accuracy was greatly improved to (97.74%). A Student's t-test shows that the improvement is statistically significant (p-values between 0.0011 and 0.0322). As shown in Table (3.6), the accuracy obtained by MSD for detecting facial expression in MMI has been improved to (99.42%) using EMSD. This difference is, once again, statistically significant according to the Student's t-test (p-values between 0.0320 and 0.0420). Moreover, EMSD is 5 and 9 times faster than the corresponding MSD for MMI and CK+ datasets, respectively.



	Message Length	
M	MMI	CK+
4	442571	482265
5	363596	390152
6	178034	202985
7	197229	273219
8	222397	293337

Figure 3.5: Message length values as a function of the clusters number for the facial expression datasets.

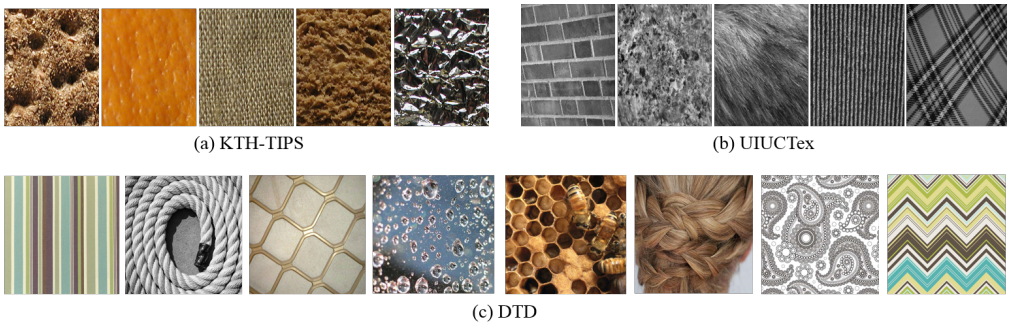
Fig. (3.5) shows that the MML criterion is capable of selecting the optimal number of clusters to represent the data. The number of classes that minimizes the message length was $M = 6$ for both datasets, which agrees with the true pre-specified number of classes.

3.6.3 Texton-based Texture Clustering

Our visual world is richly filled with a great variety of textures, present in many application areas, including industrial automation, remote sensing and biomedical image processing [119]. Classifying texture images has attracted extensive research attention for over 50 years, dating back at least to Julesz in 1962 [120]. As a classical pattern recognition problem, texture classification primarily consists of two critical subproblems: texture representation and classification. Generally, if poor features are used, even the best classifier will fail to achieve good results. Thus, the extraction of powerful texture features plays a relatively important role. Several approaches have been extensively studied with impressive performance. In this work, we used the texture analysis approach that model texture as a probabilistic process that generates small patches. The representation is obtained by means of a frequency histogram that measures how often texture patches from a codebook occur in the textured surface. The resulting representations are generally called "textons" [121, 122]. In these experiments, we consider existing natural texture image datasets that have been released and commonly used by the research community for texture classification, as summarized in Table (3.7) and sample images from each dataset are presented in Fig. (3.6).

Table 3.7: Texture image datasets considered in the experiments.

Data set	Classes	Samples per class	Total samples	Resolution (pixels)
KTH-TIPS	10	81	810	200×200
UIUCTex	25	40	1000	640×480
DTD	47	210	9870	$\sim 400 \times 400$



(a) KTH-TIPS (b) UIUCTex

(c) DTD

Figure 3.6: Examples of images from the texture datasets.

The first data set was introduced by researchers from the Royal Institute of Technology called "KTH Textures under varying Illumination, Pose, and Scale"(**KTHTIPS**) [123]. To date, this database contains ten materials: sandpaper, crumpled aluminum foil, styrofoam, sponge, corduroy, linen, cotton, brown bread, orange peel, and cracker. The **UIUC** (University of Illinois Urbana-Champaign) dataset collected by Lazebnik et al. [124] contains 25 texture classes, with each class having 40 uncalibrated, unregistered images. It has significant variations in scale and viewpoint as well as nonrigid deformations. The challenges of this database are that there are few sample images per class, but with significant variations within classes. The last considered data set is the **DTD** dataset [125], consisting of 120 texture based on a vocabulary of 47 human-interpretable texture attributes. The large intraclass variations in the DTD are different from other texture datasets, in an effort to support directly real-world applications. Evaluation results for the three datasets generated by the different approaches are summarized in Table (3.8). We start by detecting local regions and computing their SIFT descriptors [90], giving a 128-dimensional vector for each keypoint. Then, we construct a global texton vocabulary via the clustering of descriptors obtained from the different training classes. Following [126], we extract 10 textons using K-means (*i.e.*, the textons are actually the K-means cluster centers) for each texture class and then concatenate the textons of the different

Table 3.8: Texture classification accuracy using different approaches.

Dataset	MM	DCM	MGD	MBL	**MSD**	**EMSD**
KTH-TIPS	92.63	93.14	95.29	95.20	96.00	96.02
UIUCTex	95.03	96.12	97.82	97.85	97.29	98.36
DTD	91.80	92.33	94.12	94.25	94.20	94.70

classes to form the visual vocabulary. Thus, the vocabulary sizes are 100, 250, and 470 for the KTH-TIPS, UIUCTex, and DTD datasets, respectively. Clearly, the MSD, MGD, and the MBL outperform the other approaches, and a Student's t-test shows that the differences in performance are statistically significant (*i.e.*, p-values between 0.0030 and 0.037 for the different runs). The results show also that the MSD performance is comparable to that of the MGD and MBL (*i.e.*, the differences are not statistically significant, p-values are 0.1127 and 0.2060). For texture classification, we can see that accuracy achieved using EMSD is comparable to MSD for the three considered data set. This can be explained by the relatively small texture vocabulary size. In other words, the consideration of the approximation approach is justified by the challenge in clustering high-dimensional and sparse data, *i.e.,* the improvement in accuracy compared to the corresponding model, MSD is usually obtained given the high-dimensionality of the data. Moreover, it is noteworthy also that there is no significant improvement in the time complexity when using EMSD compared to MSD.

3.7 Conclusion

In this work, we have introduced the MSD, which is a composition of the multinomial and the scaled Dirichlet distributions for count data modeling. The approach proposed is motivated by the hierarchical Bayesian framework for modeling text data and can be used in many practical situations where the generated data is in the form of vectors of frequencies. The scaled Dirichlet has several convenient properties that make it more useful and practical than the Dirichlet as a prior to the multinomial: it has an extra set of parameters that allows more modeling flexibility. For learning a finite mixture of MSD, the Expectation-Maximization (EM) algorithm has been outlined. Furthermore, we have introduced a new family of distributions (EMSD) based on the exponential family approximation of the proposed Multinomial scaled Dirichlet (MSD) to cluster high-dimensional

sparse count data faster and more efficient. The deterministic annealing expectation-maximization (DAEM) algorithm has been proposed to estimate the parameters of the EMSD mixture model, where the number of components is selected using the presented MML-based criterion.

The effectiveness of both new mixtures was shown through extensive experiments on challenging clustering problems such as text document classification, facial expression recognition, natural scene categorization, and texture classification. Results revealed that MSD mostly outperforms the mixtures of Multinomial and DCM, and achieve comparable performance compared to the recently introduced MGD and MBL. On the other hand, EMSD successfully and correctly captures the burstiness phenomenon while being many times faster and computationally efficient compared to the corresponding MSD. Our unsupervised algorithm provides promising results in selecting the optimal number of clusters by optimizing the message length of the data efficiently.

Appendix 1: Proof of Eq.(3.5)- The Marginal Distribution MSD

We have:

$$\begin{aligned}
\mathcal{MSD}(\mathbf{X}|\boldsymbol{\alpha},\boldsymbol{\beta}) &= \int_{\rho} \mathcal{M}(\mathbf{X}|\boldsymbol{\rho})\mathcal{SD}(\boldsymbol{\rho}|\boldsymbol{\theta})d\rho \\
&= \int_{\rho} \frac{n!}{\prod_{w=1}^{W} x_w!} \prod_{w=1}^{W} \rho_w^{x_w} \frac{\Gamma(a)}{\prod_{w=1}^{W}\Gamma(\alpha_w)} \frac{\prod_{w=1}^{W} \beta_w^{\alpha_w}\rho_w^{\alpha_w-1}}{\left(\sum_{w=1}^{W}\beta_w\rho_w\right)^a} d\rho \\
&= \frac{n!}{\prod_{w=1}^{W} x_w!} \frac{\Gamma(a)}{\prod_{w=1}^{W}\Gamma(\alpha_w)} \prod_{w=1}^{W} \beta_w^{\alpha_w} \int_{\rho} \frac{\prod_{w=1}^{W} \rho_w^{x_w+\alpha_w-1}}{\left(\sum_{w=1}^{W}\beta_w\rho_w\right)^a} d\rho
\end{aligned} \tag{3.25}$$

Using the fact that the integration of the pdf=1, we have: $\int_\rho \mathcal{SD}(\rho|\alpha,\beta)d\rho = 1$, straightforward manipulation yield:

$$\int_\rho \frac{\Gamma(a)}{\prod_{w=1}^{W}\Gamma(\alpha_w)} \frac{\prod_{w=1}^{W}\beta_w^{\alpha_w}\rho_w^{\alpha_w-1}}{\left(\sum_{w=1}^{W}\beta_w\rho_w\right)^a} d\rho = 1$$
$$\frac{\Gamma(a)\prod_{w=1}^{W}\beta_w^{\alpha_w}}{\prod_{w=1}^{W}\Gamma(\alpha_w)} \int_\rho \frac{\prod_{w=1}^{W}\rho_w^{\alpha_w-1}}{\left(\sum_{w=1}^{W}\beta_w\rho_w\right)^a} d\rho = 1 \tag{3.26}$$

Empirically, we found the following approximation:

$$\left(\sum_{w=1}^{W}\beta_w\,\rho_w\right)^{\sum_{w=1}^{W}\alpha_w} \simeq \prod_{w=1}^{W}\beta_w^{\alpha_w} \tag{3.27}$$

Considering this approximation to solve the integration in Eq.(3.26), gives:

$$\int_\rho \frac{\prod_{w=1}^{W}\rho_w^{\alpha_w-1}}{\left(\sum_{w=1}^{W}\beta_w\rho_w\right)^a} d\rho = \frac{\prod_{w=1}^{W}\Gamma(\alpha_w)}{\Gamma(a)\prod_{w=1}^{W}\beta_w^{\alpha_w}} \tag{3.28}$$

Using Eq.(3.28) to solve the integration for $\mathcal{SD}(\rho|\alpha+x,\beta)$ in Eq. (3.25), we obtain:

$$\mathcal{MSD}(\mathbf{X}|\boldsymbol{\alpha},\boldsymbol{\beta}) = \frac{n!}{\prod_{w=1}^{W}x_w!}\frac{\Gamma(a)}{\prod_{w=1}^{W}\Gamma(\alpha_w)}\prod_{w=1}^{W}\beta_w^{\alpha_w} \times \frac{\prod_{w=1}^{W}\Gamma(\alpha_w+x_w)}{\Gamma(\sum_{w=1}^{W}\alpha_w+x_w)\prod_{w=1}^{W}\beta_w^{\alpha_w+x_w}}$$

$$= \frac{n!}{\prod_{w=1}^{W}x_w!}\frac{\Gamma(a)}{\Gamma(a+n)\prod_{w=1}^{W}\beta_w^{x_w}}\prod_{w=1}^{W}\frac{\Gamma(\alpha_w+x_w)}{\Gamma(\alpha_w)}$$

Appendix 2: Hessian Matrix for MSD

The gradient vector G is obtained by computing the first partial derivative of $\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)$ with respect to α_{jw} and $\beta_{jw}, w = 1, \ldots, W$, we obtain:

$$\frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial \alpha_{jw}} = \sum_{i=1}^{N} z_{ij} \Big(\Psi(a_j) - \Psi(a_j + n_i) + \Psi(\alpha_{jw} + x_{iw}) - \Psi(\alpha_{jw}) \Big) \tag{3.29}$$

$$\frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial \beta_{jw}} = \sum_{i=1}^{N} z_{ij} \left(\frac{-x_{iw}}{\beta_{jw}} \right) \tag{3.30}$$

where Ψ is the digamma function (the logarithmic derivative of the Gamma function). The Hessian matrix is based on the second-order derivatives calculated as follows:

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial \alpha_{jw1} \partial \alpha_{jw2}} = \begin{cases} \sum\limits_{i=1}^{N} z_{ij} \Big(\Psi'(a_j) - \Psi'(a_j + n_i) + \Psi'(\alpha_{jw} + x_{iw}) - \Psi'(\alpha_{jw}) \Big) & \text{if} \quad w_1 = w_2 = w, \\ \sum\limits_{i=1}^{N} z_{ij} \Big(\Psi'(a_j) - \Psi'(a_j + n_i) \Big) & \text{otherwise,} \end{cases} \tag{3.31}$$

where Ψ' is the trigamma function, and:

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial \beta_{jw1} \partial \beta_{jw2}} = \begin{cases} \sum\limits_{i=1}^{N} z_{ij} \left(\frac{x_{iw}}{\beta_{jw}^2} \right) & \text{if } w_1 = w_2 = w, \\ 0 & \text{otherwise,} \end{cases} \tag{3.32}$$

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}_j|\theta)}{\partial \alpha_{jw} \beta_{jw}} = \frac{\partial^2 \mathcal{L}(\mathcal{X}_j|\theta)}{\partial \beta_{jw} \alpha_{jw}} = 0 \tag{3.33}$$

Appendix 3: Fisher Information Matrix for EMSD

$F(\theta_j)$ is obtained by calculating the negative of the log-likelihood function given by:

$$-\mathcal{Q}(\mathcal{X}_j|\boldsymbol{\theta}_j) = \eta_j(-\log\Gamma(s_j)) + \sum_{d=l}^{l+\eta_j-1} \log\Gamma(s_j+n_d) - \sum_{w:x_{dw}\geq 1} \Big(\log\lambda_{jw} - \log x_{dw} - x_{dw}\log(\nu_{jw})\Big) \tag{3.34}$$

Then, the first order derivative with respect to λ_{jw}, and $\nu_{jw}, w = 1,\ldots,W$, also called the Fisher score function, is:

$$-\frac{\partial\mathcal{Q}(\mathcal{X},\mathcal{Z}|\Theta)}{\partial\lambda_{jw}} = \eta_j(-\Psi(s_j)) + \sum_{d=l}^{l+\eta_j-1}\Psi(s_j+n_d) - \sum_{d=l}^{l+\eta_j-1} I(x_{dw}\geq 1)\frac{1}{\lambda_{jw}} \tag{3.35}$$

$$-\frac{\partial\mathcal{Q}(\mathcal{X},\mathcal{Z}|\Theta)}{\partial\nu_{jw}} = \sum_{d=l}^{l+\eta_j-1} I(x_{dw}\geq 1)\frac{1}{\nu_{jw}} \tag{3.36}$$

where Ψ is the digamma function. Then,

$$-\frac{\partial^2\mathcal{Q}(\mathcal{X}_j|\theta)}{\partial\lambda_{jw1}\lambda_{jw2}} = \begin{cases} \eta_j(-\Psi'(s_j)) + \sum_{d=l}^{l+\eta_j-1}\Psi'(s_j+n_d) + \sum_{d=l}^{l+\eta_j-1} I(x_{dw}\geq 1)\frac{1}{\lambda_{jw}^2}, & w_1 = w_2, \\ \eta_j(-\Psi'(s_j)) + \sum_{d=l}^{l+\eta_j-1}\Psi'(s_j+n_d),\ otherwise \end{cases} \tag{3.37}$$

where Ψ' is the trigamma function, and:

$$-\frac{\partial^2\mathcal{Q}(\mathcal{X}_j|\theta)}{\partial\nu_{jw1}\nu_{jw2}} = \begin{cases} \sum_{d=l}^{l+\eta_j-1} I(x_{dw}\geq 1)\frac{-1}{\nu_{jw}^2}, & w1 = w2 \\ 0 & ,otherwise \end{cases} \tag{3.38}$$

$$-\frac{\partial^2\mathcal{Q}(\mathcal{X}_j|\theta)}{\partial\lambda_{jw}\nu_{jw}} = -\frac{\partial^2\mathcal{Q}(\mathcal{X}_j|\theta)}{\partial\nu_{jw}\lambda_{jw}} = 0 \tag{3.39}$$

Chapter 4

Hybrid Generative/Discriminative Approaches Based on Multinomial Scaled Dirichlet Mixture Models

Developing both generative and discriminative techniques for classification has achieved significant progress in the last few years. Considering the capabilities and limitations of both, hybrid generative discriminative approaches have received increasing attention. Our goal is to combine the advantages and desirable properties of generative models, *i.e.,* finite mixture, and the Support Vector Machines (SVMs) as powerful discriminative techniques for modeling count data that appears in many domains in machine learning and computer vision applications. In particular, we select accurate kernels generated from mixtures of Multinomial Scaled Dirichlet distribution and its exponential approximation (EMSD) for support vector machines. We demonstrate the effectiveness and the merits of the proposed framework through challenging real-world applications, namely; object recognition and visual scenes classification. Large scale datasets have been considered in the empirical study such as Microsoft MOCR, Fruits-360, and MIT places.

4.1 Introduction

The different approaches to manage, filter and retrieve information can be grouped into two main categories of approaches: model-based (generative) approaches and discriminative classifiers. The goal of a generative model is to estimate the class-conditional distributions $P(\mathbf{X}|j)$ for $j = 1, \ldots, M$, where M is the total number of classes, and the prior probabilities (*i.e.,* mixing weights) p_j of each class, which are then used for classification via Bayes' rule [127]. Examples include Hidden Markov Models, Bayesian Networks, mixture models, etc. On the other hand, discriminative approaches focus directly on the classification problem (*i.e.,* the problem of primary interest) by estimating a classification function $j = f(\mathbf{X})$ directly from the data without regard to the underlying class densities [128]. Discriminative classifiers, such as support vector machines and neural networks, generally have superior classification performance [129]. However, when the amount of available labeled training data is small, generative approaches may provide better classification results by providing a principled framework for handling uncertainty and missing data [130]. In many settings, traditional generative or discriminative methods either are infeasible or fail to provide acceptable results and generalization to new data. Instead, researchers have recently turned to hybrid generative discriminative approaches that can efficiently combine the advantages of both approaches and then get the best of both worlds [131]. Hybrid approaches can be viewed actually as the incorporation of prior knowledge about the problem at hand into the training procedure to obtain the best possible performance [132, 133].

This paper is an extended version of our earlier work based on a novel mixture model that we called the Multinomial Scaled Dirichlet (MSD) [23]. The proposed model is the composition of the Scaled Dirichlet distribution and the Multinomial in the same way that the DCM [9], MGD [10], MBL [11] are the compositions of the Dirichlet, the generalized Dirichlet, and the Beta-Liouville, respectively, with the Multinomial. The Scaled Dirichlet is a generalization of the Dirichlet distribution, which is the best-known distribution for categorical data modeling, and it has shown to be an interesting prior to the Multinomial. We have argued, in our previous work, that the finite mixture of MSD distributions is a more appropriate and flexible generative model than the best state-of-the-art methods for text data. Moreover, MSD could capture the burstiness phenomenon, *i.e.*, if a

word appears once it is much more likely to appear again. Indeed, this phenomenon translates to images represented by a bag-of-features (BOF), as visual elements tend, also, to appear in bursts [96]. Thus, we extended the previous work to model visual data where burstiness is also important. Here, we first provide a close approximation to the MSD as a member of the exponential family of distributions that we called EMSD. Then, we further extend the work by proposing a hybrid model devoted to the applications in which count data representations are involved. Several well-motivated SVM kernels have been developed based on MSD/EMSD mixture models. In particular, we develop a Fisher kernel between two MSD/EMSD distributions and closed-form expressions of different information-divergence based kernels, namely, Kullback–Leibler kernel, Reńyi kernel, and Jensen–Shannon kernel.

The remainder of the paper is organized as follows. In Section 4.2, we briefly review generative, discriminative, and hybrid approaches. In Section 4.3, we present the Multinomial Scaled Dirichlet mixture model and Expectation-Maximization (EM) algorithm for learning its parameters and provide the exponential family approximation to the MSD which we call EMSD. The derivation of different SVM kernels from a mixture of MSD/EMSD distributions is discussed in Section 4.4. Section 4.5 is devoted to the experimental results. Finally, Section 4.6 concludes the paper.

4.2 Related Works

4.2.1 Generative Models for Count Data

The Multinomial models are simple and convenient to use, thus, they are very popular and often serve as basic units in complex models. Multinomial distribution, the multivariate generalization of the Binomial distribution, is widely used in modeling categorical data Consider that we have a set of N observations $\mathcal{X} = \{X_1, \ldots, X_N\}$, where each vector $\mathbf{X}_i$ represents a given image (or document) and is described in terms of the counts (or frequencies) of D features, $\mathbf{X}_i = (x_{i1}, \ldots, x_{iD})$, where the features could be, for instance, words in the case of textual documents or visual words in the case of images. Then, the probability of an object $\mathbf{X}$, represented as a vector of counts, that it follows a

Multinomial distribution with parameters $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_D)$, is given by:

$$\mathcal{M}(\mathbf{X}|\boldsymbol{\rho}) = \frac{n!}{\prod_{d=1}^{D} x_d!} \prod_{d=1}^{D} \rho_d^{x_d} \tag{4.1}$$

where D is the size of the vocabulary, and $n = \sum_{d=1}^{D} x_d$ is the document length.

The main drawback of Multinomial models is that they make a naive Bayes assumption: that the probability of each word event in a document is independent of the words context and its position in the document [52], which is not very accurate. In natural languages, the word frequencies have been shown to be affected by the phenomenon of burstiness [7, 54]. Thus, modeling the probabilities of repeat occurrences of words improves the classification performance and information retrieval accuracy. Indeed, Multinomial distributions fail to capture this phenomenon well, as was shown in [9]. An alternative approach for modeling term frequencies is the Dirichlet Compound Multinomial (DCM) proposed in [9], where the authors proved that the performance of DCM is comparable to that obtained with multiple heuristic changes to the Multinomial model. The superior performance of DCM is obtained by the hierarchical approach, which introduces the prior information into the construction of the statistical model, where the Dirichlet is generally taken as a prior to the Multinomial distribution. The Dirichlet distribution, with a set of parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_D)$, is defined as:

$$\mathcal{D}(\boldsymbol{\rho}|\boldsymbol{\alpha}) = \frac{\Gamma(A)}{\prod_{d=1}^{D} \Gamma(\alpha_d)} \prod_{d=1}^{D} \rho_d^{\alpha_d - 1} \tag{4.2}$$

where $A = \sum_{d=1}^{D} \alpha_d$. Then, the DCM is the marginal distribution given by the following integration:

$$\begin{aligned} \mathcal{DCM}(\mathbf{X}|\boldsymbol{\alpha}) &= \int_{\rho} \mathcal{M}(\mathbf{X}|\boldsymbol{\rho}) \mathcal{D}(\boldsymbol{\rho}|\boldsymbol{\alpha}) d\rho \\ &= \frac{n!}{\prod_{d=1}^{D} (x_d)!} \frac{\Gamma(A)}{\Gamma(\sum_{d=1}^{D} x_d + \alpha_d)} \prod_{d=1}^{D} \frac{\Gamma(x_d + \alpha_d)}{\Gamma(\alpha_d)} \end{aligned} \tag{4.3}$$

We can note that compared to the Multinomial, the DCM has one extra degree of freedom, since its

parameters are not constrained to sum up to one, which makes it more practical [61, 63]. Although the Dirichlet distribution is a natural conjugate prior for the Multinomial likelihood and it exhibits many convenient mathematical properties, it is not the most appropriate solution. Hence, other distributions were proposed in the literature to be used as a prior for the Multinomial. Dirichlet distribution, for instance, has a very restrictive negative covariance structure and the variables with the same mean must have the same variance [98, 100]. These properties make its use as a prior in the case of positively correlated data inappropriate. Recent works show that the generalized Dirichlet has many convenient properties that make it more useful and practical, as a prior to the Multinomial than the Dirichlet in real-life applications [10, 100]. Bouguila [11] later introduced another alternative based on Liouville family distributions, which is in contrast with the Dirichlet and, like the generalized Dirichlet, can have positive or negative covariance. In addition, like the Dirichlet and the generalized Dirichlet, the Liouville distribution, of the second kind, is a conjugate to the Multinomial distribution [100]. Other interesting properties of the Liouville distribution are discussed in [101, 102].

4.2.2 The Generative/Discriminative Learning Approach

Support Vector Machines (SVMs), as a type of classifiers, are well known for supervised learning and applicable to both classification and regression problems. Since the SVM classifier was introduced in [129], it gained popularity due to its good generalization, global solution, number of tuning parameters, and their solid theoretical foundation. The development of efficient SVMs implementations led to broadening its applications [134–136]. A challenging problem in the case of SVMs is the choice of the kernel function, which is actually a measure of similarity between two vectors. In case the data are not linearly separable, it can be mapped into a high dimensional feature space using a kernel function to simplify the computation of the inner product value of the transformed data in the feature space [137, 138]. The generally used kernel functions are polynomial, Radial Basis Function (RBF), and sigmoid [139, 140]. Given their good discrimination and generalization capabilities, SVMs are well known powerful tools for pattern classification. While the generative models (*e.g.,* mixture models and hidden Markov models) aim to estimate the class-conditional distributions, the discriminative approaches focus directly on the classification problem

by estimating a classification function. In most of the applications, it was shown that the classic SVM kernels are not the best choice, and better results can be achieved when the kernel function is generated directly from data. One of the most successful approaches is the Fisher kernel initially proposed in [141], and the main idea is to exploit the geometric structure on the statistical manifold by mapping a given individual sequence of vectors into a single feature vector, defined in the gradient log-likelihood space. The fisher kernel has been widely used in the literature. For instance, in [142] where Gaussian mixture model-based kernel functions used for speech emotion recognition, in [143, 144] based on a mixture of Dirichlet and generalized Dirichlet for modeling non-Gaussian data, and in [145] where Fisher kernels extract discriminative embeddings from Hidden Markov Models (HMMs) of occurrences for quantified self activities and behavior. Moreover, Fisher kernels have been used and shown excellent performance in many applications that involve discrete data such as handwriting recognition, speech recognition, facial expression analysis, and bioinformatics based on mixture of Multinomials [146], as well as, spam and text categorization and hierarchical classification of vacation images based on mixture of Multinomial Dirichlet distributions [147].

An alternative to the Fisher kernel is to generate SVM kernels based on information divergence between distributions. As a similarity measure between input vectors, a given kernel should capture the intrinsic properties of the data to classify, and take into account prior knowledge of the problem domain. In particular, it is a group of kernels obtained by exponentiating divergence measure between $p(\mathbf{X}|\Theta)$ and $p'(\mathbf{X}|\Theta')$. Several information divergence-based kernels have been previously proposed. For instance, the authors in [148, 149] have derived a kernel distance based on the symmetric Kullback-Leibler (KL) divergence [150] between Dirichlet distributions [151], and between Gaussian mixtures which was applied successfully for speaker identification, image classification and visual recognition [148]. Moreover, authors in [143, 152] have derived other kernel distances between Liouville mixtures and Langevin distributions, respectively, based on Renyi and Jensen–Shannon Kernels [153, 154]. It is noteworthy that the existence of closed-form expressions for these distances is demonstrated by exploiting the fact that a distribution belongs to the exponential family of distributions, such as Gaussian [155, 156], Beta-Liouville [157] and the Dirichlet family [143, 158].

4.3 Finite Multinomial Scaled Dirichlet mixture model

In this section, we first present, in sufficient detail, the Multinomial Scaled Dirichlet mixture model and EM algorithm for learning its parameters as previously proposed in [23]. Then, we derive a new distribution that is an approximation to the MSD, which we call EMSD, and it is, unlike the MSD, a member of the exponential family.

4.3.1 Multinomial Scaled Dirichlet (MSD)

In this work, we look at some limitations of the Dirichlet distribution. For instance, Dirichlet does not take into account relative positions between categories or Multinomial cells [98]. Moreover, it has a poor parameterization that limits its ability to better model variance and covariance [97]. Thus, we are proposing the choice of a more flexible prior to the Multinomial that can help to resolve these issues, which is a generalization of the Dirichlet called Scaled Dirichlet distribution [98, 103]. The Scaled Dirichlet is a generalization of the Dirichlet distribution, which is the distribution of a random vector obtained after applying the perturbation and powering operations to a Dirichlet random composition. These operations define a vector-space structure in the simplex and play the same role as sum and product by scalars in real space [103].

The Scaled Dirichlet has a set of parameters; $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_D)$ which is the shape parameter, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_D)$ which is the scale parameter. The probability density of the probability vector $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_D)$, is given by [103]:

$$\mathcal{SD}(\boldsymbol{\rho}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\Gamma(A)}{\prod_{d=1}^{D} \Gamma(\alpha_d)} \frac{\prod_{d=1}^{D} \beta_d^{\alpha_d} \rho_d^{\alpha_d - 1}}{\left(\sum_{d=1}^{D} \beta_d \rho_d \right)^A} \tag{4.4}$$

where $A = \sum_{d=1}^{D} \alpha_d$, and Γ denotes the Gamma function. The shape parameter $\boldsymbol{\alpha}$ simply describes the form or shape of the Scaled Dirichlet distribution, and its flexibility is very significant in finding patterns and shapes inherent in a data set. The scale parameter $\boldsymbol{\beta}$ controls how the density plot is spread out where the shape of the density is invariant, irrespective of the value of a constant or uniform scale parameter. Note that the Dirichlet distribution is a special case of the scaled Dirichlet

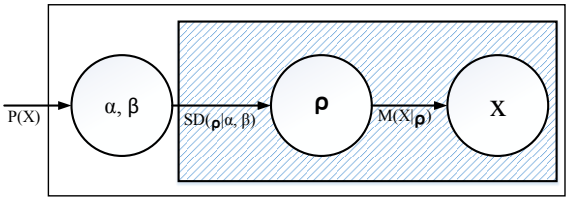


Figure 4.1: Hierarchical representation of MSD model.

that can be obtained when all elements of the vector $\boldsymbol{\beta}$ are equal to a common constant. Thus, the Scaled Dirichlet includes the Dirichlet as a special case. Compared to the Dirichlet, the Scaled Dirichlet, has D extra parameters, which enhances the model flexibility [106, 107].

Integrating over ρ gives the marginal distribution of $\mathbf{X}$, as follows:

$$\begin{aligned} \mathcal{MSD}(\mathbf{X}|\boldsymbol{\alpha},\boldsymbol{\beta}) &= \int_{\rho} \mathcal{M}(\mathbf{X}|\boldsymbol{\rho})\mathcal{SD}(\boldsymbol{\rho}|\boldsymbol{\alpha},\boldsymbol{\beta})d\rho \\ &= \frac{n!}{\prod_{d=1}^{D} x_d!} \frac{\Gamma(A)}{\Gamma(n+A)} \frac{}{\prod_{d=1}^{D} \beta_d^{x_d}} \prod_{d=1}^{D} \frac{\Gamma(x_d+\alpha_d)}{\Gamma(\alpha_d)} \end{aligned} \tag{4.5}$$

The last step of equation (4.5) is obtained by using the fact that $\int_{\rho} \mathcal{SD}(\boldsymbol{\rho}|\boldsymbol{\alpha},\boldsymbol{\beta}) = 1$, and applying the following empirically tested approximation:
$\left(\sum_{d=1}^{D} \beta_d\, \rho_d\right)^{\sum_{d=1}^{D} x_d} \simeq \prod_{d=1}^{D} \beta_d^{x_d}$, given a common constant value for β. By setting $\beta_1 = \beta_2 = \cdots = \beta_D = 1$, equation (4.5) is reduced to (4.3), which is the DCM.

In Figure (4.1), we present the graphical representation of the MSD model. Like DCM, MSD is a hierarchical Bayesian modeling framework that can be interpreted as bag-of-bags-of words, where for a specific document, for instance, the Multinomial is linked to particular sub-topics, and thus, it makes the emission of some words more likely than others. This gives it the ability to handle *burstiness* even for rare words without introducing heuristics [61]. That is, if a rare word appears once in a document, it is much more likely to appear again.

4.3.2 MSD Mixture Learning

In mixture modeling, we assume that our data population is generated from a mixture of sub populations. Given an observed data set $\mathcal{X}$ with N data instances $\{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, each D-dimensional vector representing an object $\mathbf{X}_i = (x_{i1}, \ldots, x_{iD})$, is drawn from a superposition of K Multinomial

Scaled Dirichlet densities of the form:

$$p(\mathbf{X}_i|\boldsymbol{\pi},\boldsymbol{\alpha},\boldsymbol{\beta}) = \sum_{k=1}^{K} \pi_k \, \mathcal{MSD}(\mathbf{X}_i|\boldsymbol{\alpha}_k,\boldsymbol{\beta}_k) \tag{4.6}$$

where π_k ($0 < \pi_k < 1$ and $\sum_{k=1}^{K} \pi_k = 1$) are the mixing proportions. Next, we introduce a K-dimensional binary random vector $\mathbf{Z_i} = (z_{i1}, \ldots, z_{iK})$ to each data vector $\mathbf{X_i}$, where $z_{ik} \in \{0, 1\}$ and $\sum_{k=1}^{K} z_{ik} = 1$. Here, the latent variable $\mathbf{Z_i}$ works as an indicator variable equals to 1 if $\mathbf{X_i}$ belongs to component k and 0 otherwise.

The complete data at this case are $(\mathcal{X}, \mathcal{Z}|\Theta)$, where $\mathcal{X}$ represents a set of observed variables, and the set $\Theta = (\alpha_1, \ldots, \alpha_K, \beta_1, \ldots, \beta_K, \pi_1, \ldots, \pi_K)$ denotes all latent variables and parameters. For learning a mixture model, Expectation Maximization (EM) algorithm can be used to obtain the maximum likelihood estimates of the parameters [160].

In the ***E-step*** of the EM algorithm, we compute the posterior probabilities (*i.e.,* the probability that a vector $\mathbf{X}_i$ belongs to cluster k), as:

$$\hat{z}_{ik} = p(k|\mathbf{X}_i,\theta_k) = \frac{\pi_k \, p(\mathbf{X}_i|\theta_k)}{\sum_{k=1}^{K} \pi_k \, p(\mathbf{X}_i|\theta_k)} \tag{4.7}$$

In the ***M-step***, we update the model parameter estimates according to:

$$\begin{aligned} \Theta^* &= \arg \max_{\Theta} \{\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)\} \\ &= \arg \max_{\Theta} \sum_{i=1}^{N} \sum_{k=1}^{K} \hat{z}_{ik} \, \log \left(p(\mathbf{X}_i|\theta_k)\pi_k\right) \end{aligned} \tag{4.8}$$

when maximizing (4.8) we obtain:

$$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \hat{z}_{ik} \tag{4.9}$$

We can obtain the maximum likelihood parameter estimates for the MSD by taking the derivative of the log-likelihood function and find Θ_{MLE} when the derivative is equal to zero. However, we do not obtain a closed-form solution for the $\boldsymbol{\alpha_k}$ and $\boldsymbol{\beta_k}$ parameters. We, therefore, use the

Algorithm 4: EM for estimating the MSD mixture parameters.

Output: Optimal parameters θ^*
Input: Dataset $\mathcal{X}$ with N D-dimensional vectors, a specified number of clusters K

```
Initalization: Apply k-means on the ND-dimensional vectors to obtain initial K clusters ;
Initialize the shape parameters α_k using method of moments ;
Initialize the scale parameter vector β_k with a vector of ones;
repeat
The Expectation Step(E-step);
for i = 1 to N do
    for k = 1 to K do
        Compute the posterior probabilities p(k|X_i, θ_k) using equation (4.7);
    end
end
for k = 1 to K do
    Update the mixing proportion π_k using equation(4.9) ;
    Update the θ_k using Equation (4.10);
end
```

Newton-Raphson method expressed as:

$$\theta_k^{new} = \theta_k^{old} - H^{-1}G \tag{4.10}$$

where H is the Hessian matrix associated with the complete log-likelihood $\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)$, and G is the first derivatives vector. To calculate the Hessian matrix, we have to compute the second and mixed derivatives of the log-likelihood function. The complete block Hessian matrix H_k has to be transformed to its inverse before it can be used in the Newton-Raphson maximization. To achieve an optimal performance, a proper initialization is needed to avoid converging to local maxima. To initialize the π_k parameter, we use the K-means algorithm, and to initialize the model parameters, we make use of the method of moments.

In the case of the Multinomial Scaled Dirichlet distribution, a closed-form solution for its moment equations does not exist. Thus, we will initialize the $\boldsymbol{\alpha}_k$ vector using the moments' equations of the DCM distribution [75], while the $\boldsymbol{\beta}_k$ vector will be initialized with equal scaling (a vector of ones). Parameters will be then updated during the EM iterations to take their natural values in relation to the observed data. The complete algorithm for learning the MSD mixture parameters is summarized in (Algorithm 4).

4.3.3 MSD Approximation to the Exponential Family

Any family of distributions where the support depends on the parameter can not be from an exponential family. However, it can always be reduced to a member of the exponential families via a suitable transformation and re-parameterization. A multi-parameter exponential distribution for random variables $\mathbf{X}$ indexed by a parameters set θ, can be written as [67]:

$$P(\mathbf{X}|\theta) = H(X)\exp\{G(\theta)T(X) + \Phi(\theta)\} \tag{4.11}$$

where $G(\theta)$ is called the natural parameter, $T(X)$ is the sufficient statistic, $H(X)$ is the underlying measure, and $\Phi(\theta)$ is called log normalizer which ensures that the distribution integrates to one.

Given the sparsity nature of data sets represented using bag-of-words, or bag-of-visual-words, it should be possible to evaluate the probability as a function of non-zero x_d values only for computational efficiency. That is, the value of $x_d! = 1$, $\beta_d^{x_d} = 1$ and $\Gamma(\alpha_d + x_d)/\Gamma(\alpha_d) = 1$ when $x_d = 0$. The MSD distribution in this case is given by:

$$\mathcal{MSD}(\mathbf{X}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{n!}{\prod\limits_{d:x_d\geq 1} x_d!}\frac{\Gamma(A)}{\Gamma(A+n)}\frac{}{\prod\limits_{d:x_d\geq 1}\beta_d^{x_d}}\prod_{d:x_d\geq 1}\frac{\Gamma(\alpha_d + x_d)}{\Gamma(\alpha_d)} \tag{4.12}$$

In case of high dimensional data when the parameters are really small, we can use the approximation given in [12] to replace $\Gamma(\alpha_d + x_d)/\Gamma(\alpha_d)$ by $\Gamma(x_d)\alpha_d$ in the previous equation. Using the fact that if x is an integer then $x! = x(x-1)!$, we can further simplify to obtain the new distribution that we call (EMSD):

$$\mathcal{EMSD}(\mathbf{X}) = \frac{n!}{\prod\limits_{d:x_d\geq 1} x_d}\frac{\Gamma(s)}{\Gamma(s+n)}\prod_{d:x_d\geq 1}\frac{\lambda_d}{\nu_d^{x_d}} \tag{4.13}$$

where $s = \sum_{d=1}^{D}\lambda_d$. We denote the EMSD parameters as λ_d instead of α_d, and ν_d instead of β_d to distinguish them from the MSD parameters for clarity. We can rewrite Eq.(4.13) in exponential

family form as:

$$\mathcal{EMSD}(\mathbf{X}) = \Big(\prod_{d:x_d \geq 1} x_d \Big)^{-1} n! \frac{\Gamma(s)}{\Gamma(s+n)} \times \exp\Big[\sum_{d=1}^{D} I(x_d \geq 1) \log(\lambda_d) - x_d \log(\nu_d) \Big] \tag{4.14}$$

where $I(x_d \geq 1)$ is an indicator that represents whether the word d appears at least once in the vector $\mathbf{X}$.

4.4 A Hybrid of MSD/EMSD Mixture Models and SVM

The motivation of using SVMs for classification problems is well documented and discussed in [129, 161]. An important issue in applying SVMs is the choice of the kernel function, $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$, for non-separable data [162]. The idea is to capture the intrinsic properties of the data to classify based on a similarity measure between input vectors taking into account a prior knowledge of the problem domain. In this section, we develop kernels based on MSD/EMSD mixture models that address cretin practical limitations of classical kernels and could also be called generative kernels [163]. Fig. (4.2) shows the proposed hybrid generative/discriminative process graphically. A generative model can be used in a discriminative context by extracting Fisher Scores, or probability distances, from the generative model and converting them into a Kernel function. A kernel represents the data as a matrix of pairwise similarities, which may be used for classification by a kernel method, such as the support vector machine. We show the capability of the generated kernel functions in real-life applications that require handling bags of count vectors. The generative stage is done by fitting the MSD/EMSD model directly to the local SIFT feature vectors extracted from the images (*i.e.*, each image is encoded as a bag of SIFT feature vectors). Consequently, each image in our data sets is represented by a finite mixture model of MSD/EMSD distributions. The discriminative stage, on the other hand, is represented by computing the Fisher, or probability product, kernel between each of these mixture models giving us kernel matrices to feed the SVM classifier. Moreover, we have used the 1-versus-all training approach, and the values for all design parameters were obtained by performing 10-fold cross-validation.

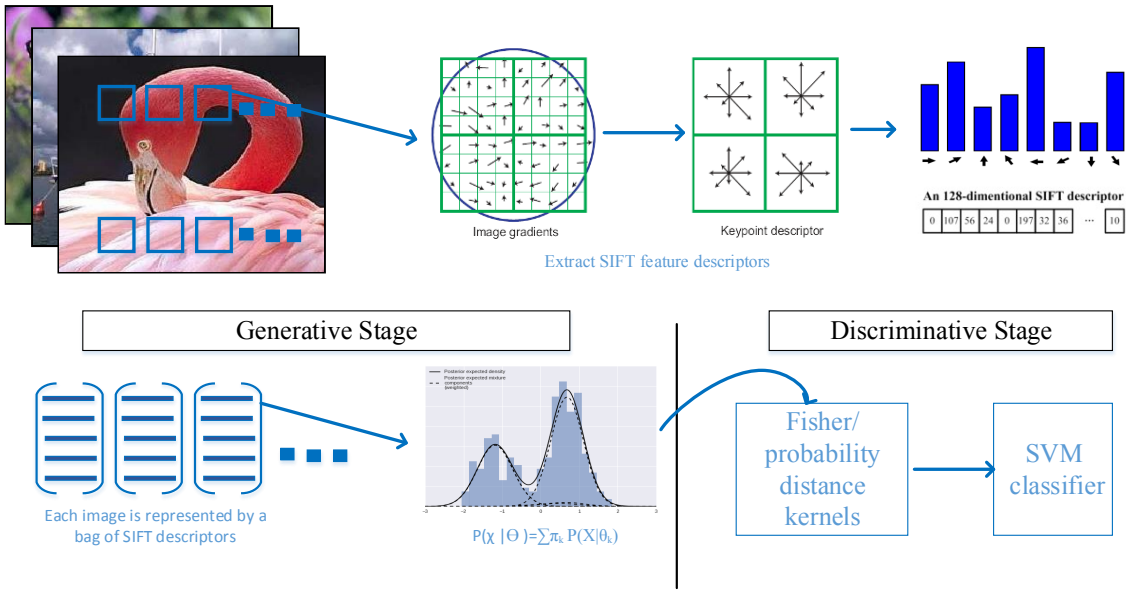


Figure 4.2: Graphical representation of the proposed hybrid learning approach.

4.4.1 Development of Fisher Kernels

Let $\mathcal{O} = \{\mathbf{O_1}, \ldots, \mathbf{O_N}\}$ be a set of multimedia objects (*e.g.,* images), where each image $\mathbf{O}_i$ is defined by a sequence of feature vectors $\mathcal{X}_{\mathbf{O_i}} = \{X_{O_i1}, \ldots, X_{O_iT}\}$. Each individual object $\mathcal{X}_{\mathbf{O_i}}$ has its own size T as the image can be represented by a bag of pixel vectors of a set of local descriptors of D dimensions [164, 165]. The Fisher kernel is defined in the gradient log-likelihood space, and the resulted feature vector is called the Fisher score and defined as $U_{\mathcal{X}_{\mathbf{O_i}}} = \frac{\partial P(\mathcal{X}_{\mathbf{O_i}}|\Theta)}{\partial \Theta}$, where each component is the derivative of the log-likelihood with respect to a particular parameter. In the case of a finite mixture model of MSD/EMSD distributions with K components, the corresponding feature space is $(K(2D+1)-1)$-dimensional. The kernel is then defined as $\mathcal{K}(\mathcal{X} : \mathcal{X}_{O_i}) = U_{\mathcal{X}} F(\Theta)^{-1} U_{\mathcal{X}_{O_i}}$, where $F(\Theta)$ is the Fisher information matrix whose role is less significant and then can be approximated by the identity matrix [141].

To simplify the notation, let $\mathcal{X} = \{X_1, \ldots, X_T\}$, each is a D-dimensional vector[1], be our sequence of feature vectors assumed to be generated by a finite mixture model with K components, so the log-likelihood for all the sequences in an object is defined as $\log P(\mathcal{X}_{O_i}|\Theta) = \prod_{t=1}^{T} P(\mathbf{X}_t|\Theta)$. By computing the gradient of $\log P(\mathcal{X}|\Theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} \hat{z}_{ik} \log \left(p(\mathbf{X}_i|\theta_k)\pi_k\right)$ with respect to the MSD

[1]The size of each vector depends on the image representation approach, in our case the vectors are 128-dimensional given that we are representing each image as a bag of SIFT descriptors [90].

model parameters, straightforward manipulations give:

$$\frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial \alpha_{kd}} = \sum_{t=1}^{T} \hat{z}_{tk} \Big(\Psi(A) - \Psi(n_t + A) + \Psi(x_{td} + \alpha_{kd}) - \Psi(\alpha_{kd}) \Big)$$
$$\frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial \beta_{kd}} = \sum_{t=1}^{T} \hat{z}_{tk} \left(\frac{-x_{td}}{\beta_{kd}} \right) \tag{4.15}$$

and with respect to the EMSD model parameters as:

$$\frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial \lambda_{kd}} = \frac{\sum_{t=1}^{T} z_{tk} I(x_{td} \geq 1)}{\sum_{t=1}^{T} z_{tk} \Psi(s_k + n_t) - M\Psi(s_k)}$$
$$\frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta)}{\partial \nu_{kd}} = -\sum_{t=1}^{T} z_{tk} I(x_{td} \geq 1) \tag{4.16}$$

where $M = \sum_{t=1}^{T} z_{tk}$ is the sum of posterior probability. Furthermore, computing the gradient $\pi_k, k = 1, \ldots, K$, which is the same for any mixture model, gives:

$$\frac{\partial \mathcal{L}(\mathcal{X}|\Theta)}{\partial \pi_k} = \sum_{t=1}^{T} \left[\frac{z_{tk}}{\pi_k} - \frac{z_{tk}}{\pi_1} \right], \quad k = 2, \ldots, K \tag{4.17}$$

Considering the unity constraint on mixing weights, we have only $K - 1$ free parameters, which explains the fact that the previous gradient equation is defined for $k \geq 2$ as π_1 can be determined knowing the values of the other mixing parameters ($\pi_1 = 1 - \sum_{k=2}^{K} \pi_k$).

4.4.2 Kernels Based on Information Divergence

The main idea of information divergence kernel is to replace the kernel computation in the original sequence space by computation in the probability density functions (PDFs) space (*i.e.,* the kernel becomes a measure of similarity between probability distributions) [149, 166]. Let $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ and $\mathcal{X}' = \{\mathbf{X}'_1, \ldots, \mathbf{X}'_N\}$ be two sequences of feature vectors representing two multimedia objects O and O', respectively, defined on the space Ω (Ω is the D-dimensional simplex in the case of EMSD distribution). The goal of this section is to generate SVM kernels based on information divergence to handle the classification of high dimensional positive vectors

for which the classic widely used kernels are not the best. The symmetric Kullback–Leibler divergence between $p(\mathbf{X}|\Theta)$ and $p'(\mathbf{X}|\Theta')$ is given by:

$$\mathcal{K}_{KL}(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) = \exp[-A\ J(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta'))] \tag{4.18}$$

where A is a kernel parameter included for numerical stability, and

$$J(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) = KL(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) + KL(p'(\mathbf{X}|\Theta'), p(\mathbf{X}|\Theta))$$

The KL divergence has a closed-form expression in the case of the EMSD distribution and is given by (see Appendix 1):

$$\begin{aligned} KL(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) = \log & \left[\frac{\Gamma\left(\sum_{d=1}^{D} \lambda_d\right).\Gamma\left(\sum_{d=1}^{D} \lambda'_d + n\right)}{\Gamma\left(\sum_{d=1}^{D} \lambda'_d\right).\Gamma\left(\sum_{d=1}^{D} \lambda_d + n\right)} \right] \\ & + \sum_{d=1}^{D} \left(\Psi(\sum_{d=1}^{D} \lambda_d + n) - \Psi(\sum_{d=1}^{D} \lambda_d) \right) (\lambda_d - \lambda'_d) \end{aligned} \tag{4.19}$$

The Reńyi kernel is another approach, based on the symmetric Reńyi divergence [153], which has been proposed in [149]:

$$\mathcal{K}_{R}(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) = \exp[-A\ R(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta'))] \tag{4.20}$$

where:

$$\begin{aligned} R(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) &= D_\sigma(p(\mathbf{X}|\Theta) + D_\sigma p'(\mathbf{X}|\Theta')) \\ &= \frac{1}{\sigma - 1} \log \int_\Omega p(\mathbf{X}|\Theta)^\sigma p'(\mathbf{X}|\Theta')^{1-\sigma} dX \\ &+ \frac{1}{\sigma - 1} \log \int_\Omega p'(\mathbf{X}|\Theta')^\sigma p(\mathbf{X}|\Theta)^{1-\sigma} dX \end{aligned} \tag{4.21}$$

where $\sigma > 0$ and $\sigma \neq 1$ is the order of Reńyi divergence. By substituting Eq.(4.21) into Eq.(4.20), we obtain the following:

$$\mathcal{K}_R(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) = \Bigg[\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} p'(\mathbf{X}|\Theta')^{1-\sigma} dX \times \int_0^{+\infty} p'(\mathbf{X}|\Theta')^{\sigma} p(\mathbf{X}|\Theta)^{1-\sigma} dX \Bigg]^{A/(1-\sigma)} \tag{4.22}$$

In the case of an EMSD distribution, we can find a closed-form expression for the Reńyi divergence (see Appendix 2):

$$\begin{aligned} \int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} p'(\mathbf{X}|\Theta')^{1-\sigma} dX = & \Big[\frac{\Gamma(\sum_{d=1}^{D} \lambda_d)}{\Gamma(\sum_{d=1}^{D} \lambda_d + n)} \Big]^{\sigma} \Big[\frac{\Gamma(\sum_{d=1}^{D} \lambda'_d)}{\Gamma(\sum_{d=1}^{D} \lambda'_d + \sum_{d=1}^{d} x_d)} \Big]^{1-\sigma} \\ & \times \frac{\Gamma(\sum_{d=1}^{D} \lambda_d + \sum_{d=1}^{D} -\sigma x_d)}{\Gamma(\sum_{d=1}^{D} \lambda_d)} \\ & \times \frac{\Gamma(\sum_{d=1}^{D} \lambda'_d + \sum_{d=1}^{D} -x_d + \sigma x_d)}{\Gamma(\sum_{d=1}^{D} \lambda'_d)} \end{aligned} \tag{4.23}$$

It is noteworthy that this kernel can be viewed as a generalization of the KL kernel since we can show that the Reńyi divergence is equal to the KL divergence as $\sigma \to 1$ [149, 153].

The last kernel is the Jensen-Shannon (JS) Kernel, generated according to the Jensen-Shannon divergence [154], and is given by [149]:

$$\mathcal{K}_{JS}(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) = \exp[-A\ JS_{\omega}(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta'))] \tag{4.24}$$

where:

$$\begin{aligned} JS_{\omega}(p(\mathbf{X}|\Theta), p'(\mathbf{X}|\Theta')) = & H[\omega\ p(\mathbf{X}|\Theta) + (1-\omega) p'(\mathbf{X}|\Theta')] \\ & - \omega\ H[p(\mathbf{X}|\Theta)] - (1-\omega)\ H[p'(\mathbf{X}|\Theta')] \end{aligned} \tag{4.25}$$

where ω is a parameter, and:

$$H[p(\mathbf{X}|\Theta)] = -\int_0^{+\infty} p(\mathbf{X}|\Theta) \log p(\mathbf{X}|\Theta) dX \tag{4.26}$$

is the Shannon entropy and we can show, that is given by the following in the case of the EMSD distribution (see Appendix 3):

$$H[p(\mathbf{X}|\Theta)] = -\log\Gamma(\sum_{d=1}^{D}\lambda_d) + \log\Gamma(\sum_{d=1}^{D}\lambda_d + n) - \sum_{d=1}^{D}\log(\lambda_d)\Big(\Psi(\sum_{d=1}^{D}\lambda_d + n) - \Psi(\sum_{d=1}^{D}\lambda_d)\Big) \tag{4.27}$$

4.5 Experimental results

4.5.1 Object Categorization

In today's world, large amounts of digital images and videos are increasingly generated. Therefore, there is an urgent need for the development of automatic methods to analyze and index these overwhelmingly digital datasets. Given a set of training images represented as pairs $(\mathbf{X}_i, C_i)$, where $\mathbf{X}_i$ is the ith feature vector representing image i and C_i is the category of the image i. The main goal here is to learn a model that affects images to specified categories.

Our evaluation was based on four different image datasets, as follows:

(1) **Caltech 101** [167] has 101 assorted object categories collected using Google Image Search. We use a subset contains $6,431$ images divided into 4 clusters. A sample of this dataset is shown in Figure (4.3).

(2) Extended-Brown **ETHZ** [168] consists of five diverse object classes: bottles, swans, mugs, giraffes, and apple logos (see Figure 4.3) with over 255 test images covering several kinds of scenes. In total, the objects appear 289 times, as some images contain multiple instances. While most are photographs, some paintings, drawings, and computer renderings are included as well.

(3) Microsoft Object Class Recognition Image (**MOCR**) dataset [169], introduced by the Microsoft research team in Cambridge, UK, contains 240 photographs belong to nine classes: building, grass, tree, cow, sky, aeroplane, face, car, and bicycle.

Caltech (4 classes) ETHZ (5 classes)

Figure 4.3: Samples from Caltech and ETHZ datasets.

Figure 4.4: Samples from Fruits-360 dataset.

(4) **Fruits-360** [170] a new, high-quality, dataset of images containing popular fruits. Currently, the set contains $55,244$ images of 81 fruits, and it is constantly updated with images of new fruits. We use a subset of 10 categories, as shown in Figure (4.4).

4.5.1.1 Comparison of Generative and Discriminative Approaches

The objective of our first set of experiments is to show the merits of our generative models, using the so-called Bag-of-Features (BoF) approach based on the frequency of "visual words" [89] provided from a visual vocabulary which is obtained by the quantization (or histogramming) of local feature vectors computed from a set of training images. Each dataset was randomly split into two halves to construct the visual vocabulary by detecting interest points from these images and calculate their descriptors using Scale-Invariant Feature Transform (SIFT) [90], in which the gradient is computed at each pixel in a 16×16 window around the detected keypoint, and in each 4×4 quadrant, a gradient orientation histogram is formed by adding the weighted gradient value

Table 4.1: Object categorization performance obtained for the different datasets using different techniques considering the BOF approach.

Dataset	Polynomial	RBF	Sigmoid	MM	DCM	MSD	EMSD
Caltech	88.18	87.95	73.02	89.50	91.22	92.55	94.91
ETHZ	77.78	76.47	74.71	82.64	83.20	85.32	96.88
MOCR	78.12	76.66	74.06	84.39	86.91	88.24	97.62
Fruits-360	68.19	65.00	63.15	83.22	87.12	90.33	95.88

to one of eight orientation histogram resulting in 128-dimensional descriptor vector. Then, the extracted vectors were clustered by using the K-means algorithm on K visual-words. Each image in the datasets was then represented by a vector describing the frequencies of a set of visual words, provided from the constructed visual vocabulary.

A summary of the classification results, measured by the average values of the diagonal entries of the confusion matrices obtained for the different classification tasks, is shown in Table 4.1. In this table, we present results obtained by different generative models as well as by SVMs with different kernels optimized for each dataset, namely polynomial kernel, Radial Basis Function (RBF) kernel, and sigmoid kernel. We can see that both MSD and EMSD perform better than the two other generative models, which themselves perform better than SVM with classic kernels. The difference between MSD and the other approaches is statistically significant (p-values between 0.027 and 0.032). Moreover, it is noteworthy that a Student's t-test, with 95 percent confidence, shows that the difference in performance between our proposed model MSD and its exponential approximation EMSD is statistically significant (*i.e.,* p-values between 0.021 and 0.003 for the different runs).

4.5.1.2 Classification Results Using the Hybrid Approach

The second set of experiments is conducted to validate our generative/discriminative approaches, and it is an alternative to quantization, which is based on the direct modeling of the generated local feature vectors by our finite mixture models. Then, the resulted models will be used to generate kernels for classification. The results obtained when we fed SVM classifiers with different kernels generated from MSD and EMSD are shown in Table 4.2. We can see that the results vary significantly across kernels. For instance, ETHZ results are between 88.18 and 92.31%, and MOCR results are

Table 4.2: Object categorization performance comparison for the hybrid learning using different kernels.

	Caltech	ETHZ	MOCR	Fruits-360
MSD-FK	97.57	88.18	89.50	94.15
EMSD-FK	98.38	88.46	89.58	95.45
Kullback	99.48	92.31	91.67	95.45
Reńyi	87.72	92.00	88.52	88.18
Jensen-Sh.	99.53	88.62	89.45	95.15

Table 4.3: Object categorization performance obtained by fitting directly different generative models to the local descriptors.

Data set	MM	DCM	MSD	EMSD
Caltech	86.61	96.10	98.37	88.41
ETHZ	86.84	96.12	96.68	92.75
MOCR	89.84	94.54	97.29	86.66
Fruits-360	88.12	95.16	95.26	88.35

between 88.52 and 91.67%. Furthermore, we can notice that the two information divergence-based kernels Kullback–Leibler (accuracy 99.48% for Caltech and 95.45% for Fruits-360) and Jensen-Shannon (accuracy 99.53% for Caltech and 95.15% for Fruits-360) has better, or at least comparable, results than the Fisher Kernels based on MSD and EMSD. However, the Reńyi kernel has shown a slight degradation in the performance with an accuracy of 87.72% and 88.18% for Caltech and Fruits-360 datasets, respectively. In particular, the Reńyi kernel has the worst performance among the tested kernels for all the datasets except for ETHZ, where all information divergence-based kernels perform slightly better than the MSD Fisher kernel.

Moreover, we compare the hybrid models to their fully generative counterparts, where the classification was based solely on the generative models. Note that the pure discriminative approach, *i.e.,* SVM with classic kernels, cannot be applied here since each image is represented now by a set of vectors). Table 4.3 shows the results of this experiment, where we can see that the proposed model MSD outperforms the other generative models. Furthermore, our experiments revealed that EMSD generally achieved better classification results when integrated with SVM than when used directly for classification purposes (*e.g.,* average classification accuracy of Caltech using EMSD is

88.41%, and can be enhanced to 99.53% using a Jensen-Shannon generative kernel).

4.5.2 Visual Scenes Modeling and Classification

The accurate organization of images enables increased efficiency of their retrieval and browsing, which is a challenging major problem in computer vision and important requirements for information systems [171]. It has several important applications like automatic linguistic indexing, retrieval, recommendation, and object detection and recognition [172]. The problem of scenes classification involves the assignment of an unknown scene to one of several classes based on a set of visual extracted features such as color, the shape of objects, and textured patterns. The goal of the experimental study in this section is to assess the effectiveness of the Multinomial Scaled Dirichlet mixture model and its exponential approximation in classifying visual scenes. Our experiments are conducted on two datasets contain highly varying outdoor and indoor scenes. The first dataset by **Fei-Fei and Perona** [93] contains 13 categories and is only available in grayscale (see Fig. 4.5). This dataset consists of nine outdoor eight categories (360 coasts, 328 forest, 374 mountain, 410 open country, 260 highway, 308 inside of cities, 356 tall buildings, 292 streets, and 241 suburb residence) and four indoor categories (174 bedroom, 151 kitchen, 289 living room, and 216 office). The average size of each image is approximately 250×300 pixels. The second dataset is a subset of **MIT Places** (the Scene Recognition Dataset) [173]. MIT Places dataset has over 7 million labeled pictures of scenes collected using three image search engines (Google Images, Bing Images, and Flickr). The subset used consists of around $60,000$ images in four categories; two indoor airport terminal and bookstore, and two outdoor forest path and ocean (see Fig. 4.6). The average size of each image is approximately 256×256 pixels.

First, each dataset has been randomly split into 80:20 to construct the visual vocabulary, and then each image was represented by a vector of visual words frequencies to be used for validating our generative models. Table 4.4 summarizes the average accuracy for classifying visual scenes datasets using different generative and discriminative approaches. When using the MSD model, the average classification accuracy for MIT places dataset was 89.15%, which is actually better than 81.35%, and 84.91%, which were achieved when we have used the Multinomial and DCM mixtures, respectively. It is noteworthy that a Student's t-test has shown that the differences in performance

Figure 4.5: Sample images from the first dataset by Fei-Fei and Perona.

Figure 4.6: Sample images from MIT places dataset; Row1: Outdoor images, Row2: Indoor images.

between the MSD and the other generative models are statistically significant. Moreover, for both datasets, the exponential approximation of MSD has shown significantly better results than the ones achieved by MSD with an average accuracy of 93.80% and 96.46% for Fei-Fie and MIT datasets, respectively. Both MSD and EMSD perform better than the SVM with the classic kernels, as shown in Table 4.4. Then, all design parameters were obtained by performing 10-fold cross-validation for the classification problem when using the different kernels. According to Table 4.5, the average accuracies for classifying the Fei-Fei dataset, for instance, were 97.39%, 99.74%, 99.50%, 99.48% and 99.74% using MSD-FK, EMSD-FK, Kullback, Reńyi, and Jensen–Shannon, respectively. These results show that combining mixture models and SVMs through different information-divergence

Table 4.4: Visual scenes classification performance obtained for the different visual scenes datasets using different techniques considering the BOF approach.

Dataset	Polynomial	RBF	Sigmoid	MM	DCM	MSD	EMSD
Fei-Fei	86.89	86.75	78.65	79.95	84.16	87.76	93.80
MIT places	83.96	86.72	79.16	81.35	84.91	89.15	96.46

Table 4.5: Visual scenes classification performance comparison using different kernels.

	Fei-Fei	MIT places
MSD-FK	97.39	98.99
EMSD-FK	99.74	99.95
Kullback	99.50	99.32
Reńyi	99.48	99.08
Jensen-Sh.	99.74	99.52

Table 4.6: Visual scenes classification performance obtained by fitting directly different generative models to the local descriptors.

Data set	MM	DCM	MSD	EMSD
Fei-Fei	87.45	94.84	96.94	94.91
MIT places	91.95	97.29	98.13	97.27

kernels outperform the SVM Fisher kernel based on MSD (the differences are statistically significant as shown by a Student's t-test, p-values between 0.027 and 0.031). However, the differences are not significant in the different information-divergence kernels.

The last set of experiments is conducted based solely on our generative models by fitting different models to the local descriptors directly. The results of this experiment are shown in Table 4.6. According to the results in Tables 4.5 and 4.6, it is clear that hybrid models improve the classification accuracy compared to their fully generative counterparts. For instance, the accuracy of classifying the MIT dataset by fitting EMSD directly to the descriptor is 97.27% compared to 99.95% when using SVM with a kernel-based on EMSD Fisher score. Moreover, we can see that fitting directly a generative model to the local SIFT feature vectors achieve significantly better results than the quantization of these vectors as it was done before (results in Table 4.4). This can be explained and interpreted by the fact that the constructed generative kernels respect local image structure in contrast with quantization which does not take into account the spatial information (*i.e.*,, geometric information about the positions of the different key points within the histogram bins).

4.6 Conclusion

In this paper, we have developed hybrid generative/discriminative approaches for count data modeling and classification through the development of a family of SVM kernels generated from our recently proposed finite mixture of Multinomial Scaled Dirichlet distributions. These approaches are motivated by the great number of applications that involve such types of data as well as the advantages of both SVMs and finite mixture models. In particular, we have introduced a new mixture model based on the Multinomial Scaled Dirichlet (MSD) and proposed an algorithm to learn a finite mixture model based on MSD using the EM algorithm. Moreover, we have provided an exponential family approximation to the MSD to be able to find closed-form expressions for information-divergence, which is demonstrated by exploiting the fact that a distribution belongs to the exponential family of distributions. Our experiments have involved object categorization and visual scenes modeling based on a local representation of images used as input to SVMs via our developed generative mixture models. The achieved results suggest that an accurate classification of count data can be achieved by efficient learning of kernels from the available data. The results have also shown clearly that the proposed MSD, and its approximation EMSD, perform better than the widely used generative models; namely Multinomial and DCM.

Appendix 1: Proof of Eq.(4.19)- Kullback–Leibler divergence for EMSD

The KL-divergence between two exponential distributions is given by [174]:

$$KL(p(X|\Theta), p'(X|\Theta')) = \Phi(\theta) - \Phi(\theta') + [G(\theta) - G(\theta')]^{tr} E_\theta[T(X)] \tag{4.28}$$

where E_θ is the expectation with respect to $p(X|\theta)$. Moreover, we have the following [67]:

$$E_\theta[T(X)] = -\Phi'(\theta) \tag{4.29}$$

Thus, according to Eq.(4.14), we have:

$$
\begin{aligned}
E_\theta\Big[\sum_{d=1}^{D} I(x_d \geq 1)\Big] &= -\frac{\partial \Phi(\theta)}{\partial \lambda_d} = \Psi(\sum_{d=1}^{D} \lambda_d + n) - \Psi(\sum_{d=1}^{D} \lambda_d) \\
E_\theta\Big[\sum_{d=1}^{D} I(x_d \geq 1) x_d\Big] &= -\frac{\partial \Phi(\theta)}{\partial \nu_d} = 0
\end{aligned} \tag{4.30}
$$

where $n = \sum_{d=1}^{D} x_d$, and $\Psi(.)$ is the digamma function. By substituting the previous two equations into Eq.(4.28), we obtain:

$$
\begin{aligned}
KL(p(X|\Theta), p'(X|\Theta')) &= \log\left(\Gamma\Big(\sum_{d=1}^{D} \lambda_d\Big)\right) - \log\left(\Gamma\Big(\sum_{d=1}^{D} \lambda'_d\Big)\right) \\
&\quad - \log\left(\Gamma\Big(\sum_{d=1}^{D} \lambda_d + n\Big)\right) + \log\left(\Gamma\Big(\sum_{d=1}^{D} \lambda'_d + n\Big)\right) \\
&\quad + \sum_{d=1}^{D}\left(\Psi(\sum_{d=1}^{D} \lambda_d + n) - \Psi(\sum_{d=1}^{D} \lambda_d)\right)(\lambda_d - \lambda'_d) \\
&= \log\left[\frac{\Gamma\Big(\sum_{d=1}^{D} \lambda_d\Big).\Gamma\Big(\sum_{d=1}^{D} \lambda'_d + n\Big)}{\Gamma\Big(\sum_{d=1}^{D} \lambda'_d\Big).\Gamma\Big(\sum_{d=1}^{D} \lambda_d + n\Big)}\right] \\
&\quad + \sum_{d=1}^{D}\left(\Psi(\sum_{d=1}^{D} \lambda_d + n) - \Psi(\sum_{d=1}^{D} \lambda_d)\right)(\lambda_d - \lambda'_d)
\end{aligned} \tag{4.31}
$$

Appendix 2: Proof of Eq.(4.23)- Reńyi divergence for EMSD

In the case of the EMSD distribution, we can show that:

$$\begin{aligned}\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} p'(\mathbf{X}|\Theta')^{1-\sigma} dX &= \Big[\frac{\Gamma(\sum_{d=1}^{D}\lambda_d)}{\Gamma(\sum_{d=1}^{D}\lambda_d + n)}\Big]^{\sigma}\Big[\frac{\Gamma(\sum_{d=1}^{D}\lambda'_d)}{\Gamma(\sum_{d=1}^{D}\lambda'_d + \sum_{d=1}^{d} x_d)}\Big]^{1-\sigma} \\ &\times \int_0^{+\infty}\left[\frac{n!}{\prod_{d=1}^{D} x_d}\prod_{d=1}^{D}\frac{\lambda_d}{\nu_d^{x_d}}\right]^{\sigma} dX \\ &\times \int_0^{+\infty}\left[\frac{n!}{\prod_{d=1}^{D} x_d}\prod_{d=1}^{D}\frac{\lambda'_d}{\nu'^{x_d}_d}\right]^{1-\sigma} dX \\ &= \Big[\frac{\Gamma(\sum_{d=1}^{D}\lambda_d)}{\Gamma(\sum_{d=1}^{D}\lambda_d + \sum_{d=1}^{D} x_d)}\Big]^{\sigma}\Big[\frac{\Gamma(\sum_{d=1}^{D}\lambda'_d)}{\Gamma(s'+n)}\Big]^{1-\sigma} \\ &\times \int_0^{+\infty}\frac{n!}{\prod_{d=1}^{D} x_d}\prod_{d=1}^{D}\lambda_d\nu_d^{-\sigma x_d} dX \\ &\times \int_0^{+\infty}\frac{n!}{\prod_{d=1}^{D} x_d}\prod_{d=1}^{D}\lambda'_d\nu'^{-x_d+\sigma x_d}_d dX \end{aligned} \tag{4.32}$$

We have the PDF of an EMSD distribution that integrates to one which gives:

$$\int_0^{+\infty}\frac{n!}{\prod_{d=1}^{D} x_d}\prod_{d=1}^{D}\frac{\lambda_d}{\nu_d^{x_d}} dX = \frac{\Gamma(\sum_{d=1}^{D}\lambda_d + \sum_{d=1}^{D} x_d)}{\Gamma(\sum_{d=1}^{D}\lambda_d)} \tag{4.33}$$

By substituting Eq.(4.33) into Eq.(4.32), we obtain:

$$\begin{aligned}\int_0^{+\infty} p(\mathbf{X}|\Theta)^{\sigma} p'(\mathbf{X}|\Theta')^{1-\sigma} dX &= \Big[\frac{\Gamma(\sum_{d=1}^{D}\lambda_d)}{\Gamma(\sum_{d=1}^{D}\lambda_d + n)}\Big]^{\sigma}\Big[\frac{\Gamma(\sum_{d=1}^{D}\lambda'_d)}{\Gamma(\sum_{d=1}^{D}\lambda'_d + \sum_{d=1}^{d} x_d)}\Big]^{1-\sigma} \\ &\times \frac{\Gamma(\sum_{d=1}^{D}\lambda_d + \sum_{d=1}^{D} -\sigma x_d)}{\Gamma(\sum_{d=1}^{D}\lambda_d)} \\ &\times \frac{\Gamma(\sum_{d=1}^{D}\lambda'_d + \sum_{d=1}^{D} -x_d + \sigma x_d)}{\Gamma(\sum_{d=1}^{D}\lambda'_d)} \end{aligned} \tag{4.34}$$

Appendix 3: Proof of Eq.(4.27)- Jensen-Shannon (JS) Kernel for EMSD

$$
\begin{aligned}
H[p(\mathbf{X}|\Theta)] &= -\int_0^{+\infty} p(\mathbf{X}|\Theta)\log p(\mathbf{X}|\Theta)dX \\
&= -\int_0^{+\infty} p(\mathbf{X}|\Theta)\Bigg[\log\Gamma(\sum_{d=1}^{D}\lambda_d) - \log\Gamma(\sum_{d=1}^{D}\lambda_d + n) \\
&+ \sum_{d=1}^{D}\log(\lambda_d)E_\theta[I(x_d \geq 1)] - \sum_{d=1}^{D}\log(\nu_d)E_\theta[I(x_d \geq 1)x_d]\Bigg]
\end{aligned} \tag{4.35}
$$

By substituting Eq.(4.30) into the previous equation, we obtain the following:

$$
\begin{aligned}
H[p(\mathbf{X}|\Theta)] &= -\log\Gamma(\sum_{d=1}^{D}\lambda_d) + \log\Gamma(\sum_{d=1}^{D}\lambda_d + n) \\
&- \sum_{d=1}^{D}\log(\lambda_d)\Big(\Psi(\sum_{d=1}^{D}\lambda_d + n) - \Psi(\sum_{d=1}^{D}\lambda_d)\Big)
\end{aligned} \tag{4.36}
$$

Chapter 5

High-Dimensional Count Data Clustering Based on an Exponential Approximation to the Multinomial Beta-Liouville Distribution

In this paper, we propose a mixture model for high-dimensional count data clustering based on an exponential-family approximation of the Multinomial Beta-Liouville distribution, which we call EMBL. We deal simultaneously with the problems of fitting the model to observed data and selecting the number of components. The learning algorithm automatically selects the optimal number of components and avoids several drawbacks of the standard EM algorithm, including the sensitivity to initialization and possible convergence to the boundary of the parameter space. We demonstrate the effectiveness and robustness of the proposed clustering approach through a set of extensive empirical experiments that involve challenging real-world applications. The results reveal that the novel proposed model strives to achieve higher accuracy compared to the state-of-the-art generative models for count data. Furthermore, the superior performance of EMBL suggests that its flexibility and ability to address the burstiness phenomenon successfully and that it is computationally efficient, especially when dealing with sparse and high-dimensional frequency vectors.

5.1 Introduction

Clustering, the process of discovering the natural grouping of a set of objects and assigning observations sharing similar characteristics to subgroups, is a significant task in data analysis and pattern recognition that attracts great attention of scholars in the last decades[29]. Most of the clustering methods have been developed for the case of continuous data. However, count data are naturally appear in numerous fields with several applications in machine learning, and computer vision (*e.g.,*[10–12, 175]).

In this work, we derive a new family of distributions that approximates an efficient and flexible generative model for count data, namely; the Multinomial Beta-Liouville (MBL) distributions, proposed previously by [11], and we call it (EMBL). Then, we deal simultaneously with the problems of model fitting and selection. We show that the exponential family approximation to MBL can dramatically improve both the clustering accuracy and computation efficiency in high-dimensional spaces.

5.1.1 Motivations

The clustering of high-dimensional count data based on the exponential Multinomial Beta-Liouville investigated in this study is motivated by the following observations.

- Clustering count data is a challenging task due to its high-dimensionality and sparse nature.
- Exponential families of distributions offer several appealing statistical and computational properties.
- Parameters estimation and performing model selection are important features of mixture based clustering methods.
- Hybrid learning approaches combine the advantages and desirable properties of both generative and discriminative techniques.

Motivation 1. In many applications, *e.g.,* text documents clustering, or image database summarization, each document or image is represented by a vector corresponding to the appearance

frequencies of words or visual words, respectively. Usually, many features occur only once, and many more do not occur at all, as each observation contains only a small subset of the vocabulary. Thus, such data are represented as high-dimensional and sparse vectors, a few thousand dimensions with a sparsity of 95 to 99% [81]. Hierarchical Bayesian modeling frameworks, such as Dirichlet Compound Multinomial (DCM) [9] and Multinomial Beta-Liouville (MBL) [11] have shown to be competitive with the best-known clustering methods for count data, but their estimation procedures are very inefficient when the collection size is large.

Motivation 2. The exponential family of distribution has finite-sized sufficient statistics, meaning that we can compress the data into a fixed-sized summary without loss of information [13, 14]. An efficient exponential-family approximation to the DCM (EDCM) has been previously proposed by Elkan [12], and it has shown to address the burstiness phenomenon successfully and to be considerably computationally faster than DCM especially when dealing with sparse and high-dimensional vectors. The fact that MBL distribution is an attractive generative model that is more flexible than DCM motivates us to approximate it as a member of the exponential family of distributions to reduce the computation and increase the efficiency in very high-dimensional spaces.

Motivation 3. The Expectation-Maximization (EM) algorithm is a broadly applicable iterative algorithm for estimating the mixture model parameters [160, 176]. Furthermore, while the majority of model selection methods depend on testing different values of the number of components, the strategy to start with a large number of components and merge them was found to be more efficient computationally [177]. We propose to extend the method and algorithm proposed in [1] to the mixture of EMBL, which deals with fitting the mixture and model selection simultaneously. Precisely, rather than selecting one among candidates models, this method directly aims at finding the best overall model in the entire set coinciding with the message length philosophy, and it is less dependent on the initialization than the standard EM.

Motivation 4. In many settings, traditional generative or discriminative methods either are infeasible or fail to provide acceptable results and generalization to new data. Instead, hybrid generative/discriminative techniques have shown to be powerful tools that generally provide lower test errors and better accuracies than either fully generative or discriminative techniques [25, 147, 178].

We address the problem of classification, where the data consists of bags of count vectors by incorporating an efficient mixture model (*i.e.,* EMBL) into Support Vector Machines (SVMs). The idea is to capture the intrinsic properties of the data to classify, taking into account prior knowledge of the problem domain.

5.1.2 Contributions

The growing demand to handle high-dimensional and sparse datasets efficiently motivates us to propose EMBL in this study. In overall, the contribution in the proposed framework can be summarized as follows:

- We proposed an exponential approximation to MBL [11] that improves its performance and computation complexity. Moreover, it provides more flexibility for several applications than the previous model with similar approach (*i.e.,* EDCM [12]).
- We proposed a learning approach that is robust in terms of initialization and simultaneously deals with fitting the mixture model to the observed data and selecting the optimal number of components, which makes it efficient for large datasets.
- We validated the proposed learning algorithm using publicly available and widely used datasets of different real-world applications that involve high-dimensional count data.
- We build new probabilistic kernels based on information divergences and Fisher score from the proposed mixture of EMBL for Support Vector Machines (SVMs). By means of standard face recognition databases, we show that our approach outperforms the widely used discriminative approaches such as SVM with classic kernels and KNN.

5.1.3 Organization

The structure of the rest of the paper is as follows. Section 5.2 reviews prior work related to this study. Section 5.3 derives the new family of distribution (EMBL), and explains the proposed algorithm for fitting a mixture of EMBL and estimate the optimal number of components. Then, experiments over real-world applications that involve high dimensional and sparse count data, are carried out in Sections 5.4, and the conclusions are drawn in Section 5.5.

5.2 Related Works

Finite mixtures are widely acknowledged in many areas, such as pattern recognition, computer vision, and machine learning. They are flexible and powerful probabilistic model-based approach to unsupervised learning (*i.e.,* clustering) of multivariate data [176, 179]. An essential problem with these approaches is to develop a probabilistic model that represents the data well by taking into account its nature. For instance, modeling the dependency of word repetitive occurrences in a text document improves the classification performance and information retrieval accuracy. Hierarchical Bayesian modeling was proposed as an appropriate and efficient solution to address this phenomenon by introducing the Dirichlet distribution as a prior to the Multinomial, which results in the Dirichlet Compound Multinomial (DCM) [9]. The Dirichlet, however, has some drawbacks, including its very restrictive negative covariance structure, inconsiderate relations between categories, and its poor parameterization [63, 100].

Model selection is a significant aspect of mixture modeling. The majority of model selection methods that have been proposed in the literature can be generally classified, from a computational point of view, into stochastic and deterministic methods. Traditionally, deterministic methods start by obtaining a set of candidates models assumed to contain the true/optimal number of clusters. According to information theory, the optimal number of clusters K is the candidate value, which minimizes the amount of information to transmit a dataset $\mathcal{X}$ efficiently from a sender to a receiver [71]. These model selection criteria are efficient techniques and have shown to give good results with mixtures models. However, their main drawbacks include the problem that might emerge with running the EM algorithm multiple times to obtain the whole set of candidates. Moreover, they select the number of components that optimally approximate the density and not necessarily the true number of classes present in the dataset [180]. Thus, the strategy to start with a large number of components and merge them was found to be more efficient computationally [177]. A practical algorithm using this strategy was proposed in [1], starting with a very large number of components and iteratively annihilates the weak components, *i.e.,* not supported by the data, and redistributes the observations, where the termination criterion is based Minimum Message Length (MML) philosophy [69, 71].

Another important aspect is fitting finite mixture models where the standard method used is the Maximum Likelihood Estimate (MLE) through the Expectation-Maximization (EM) algorithm. EM is a broadly applicable iterative algorithm for computing maximum likelihood estimates from incomplete data with an unobserved latent variables [108, 160, 176]. The main drawback of the EM algorithm is that the multi-modal nature of the likelihood function makes it highly dependent on initialization [108]. Several time-consuming solutions, proposed in the literature, have been used solely or jointly. Examples of the common strategies include using multiple random starts and choosing the one with the highest likelihood [181, 182], and initialization by clustering algorithms, such as K-means, which itself has initialization issues [3, 181]. Moreover, the Split and Merge Expectation-Maximization (SMEM) algorithm has been proposed in [183] to overcome the local maxima problem in parameter estimation of finite mixture models. Another approach is the Deterministic Annealing (DA) that has been applied successfully with hard-clustering algorithms [184, 185]. and has shown to provide good results even with non-Gaussian mixtures (see for example; [10–12]). Furthermore, the authors in [1], and [186] have proposed different robust algorithms with respect to initialization based on the component-wise EM procedure (CEM) [187].

5.3 The proposed Model

In this section, we first briefly recall the properties of the Multinomial Beta-Liouville (MBL) Distribution; then, we present the novel approximation that we call (EMBL). Afterward, we explain in detail the proposed algorithm for selecting the number of components and fitting a mixture of EMBLs. Lastly, we discuss the properties that make our proposed model efficient for clustering high-dimensional count data.

5.3.1 Multinomial Beta-Liouville (MBL) Distribution

The Liouville family of the second kind includes the Dirichlet distribution as a special case if all variables in the Liouville random vector have the same normalized variance, and the density generator variate has a Beta distribution [100]. Choosing the Beta distribution as a generating density

resulting in which is commonly called the Beta-Liouville distribution [104]. Consider, for example, a text document, or an image, which is represented as a sequence of frequencies of words, or visual words, appearances denoted by $\mathbf{X} = (x_1, \ldots, x_{D+1})$. Like the Dirichlet, the Beta-Liouville is a conjugate prior to the multinomial distribution, however, while a Dirichlet has only one degree of freedom (by selecting the value of the shape parameter α), the two more parameters in Beta-Liouville can be used to adjust the spread of the distribution which makes it more practical and provides better modeling capabilities. The probability density function of the Multinomial Beta-Liouville (MBL) distribution with parameters $\xi = (\alpha_1, \ldots, \alpha_D, \alpha, \beta)$, proposed in [11], is given by:

$$\begin{aligned} \mathcal{MBL}(\mathbf{X}|\xi) = & \frac{\Gamma\left((\sum_{d=1}^{D+1} x_d) + 1\right)}{\prod_{d=1}^{D+1} \Gamma(x_d + 1)} \\ & \times \frac{\Gamma(\sum_{d=1}^{D} \alpha_d)\Gamma(\alpha + \beta)\Gamma(\alpha')\Gamma(\beta') \prod_{d=1}^{D} \Gamma(\alpha'_d)}{\Gamma(\sum_{d=1}^{D} \alpha'_d)\Gamma(\alpha' + \beta')\Gamma(\alpha)\Gamma(\beta) \prod_{d=1}^{D} \Gamma(\alpha_d)} \end{aligned} \tag{5.1}$$

where $\alpha'_d = \alpha_d + x_d$, $\alpha' = \alpha + \sum_{d=1}^{D} x_d$, and $\beta' = \beta + x_{D+1}$.

Indeed, MBL has shown to achieve high clustering accuracy, comparably to Multinomial Scaled Dirichlet (MSD) [23], and Multinomial Generalized Dirichlet (MGD) [10], as well as it outperforms other widely used mixture models, such as mixtures of Multinomials and Dirichlet Compound Multinomial (DCM) [9]. However, MBL does not belong to the exponential family, and it is not efficient in high-dimensional spaces where many parameters need to be estimated. Inspired by the superior performance of the exponential approximation to DCM (EDCM) [12], in terms of both accuracy and computational efficiency, this work aims to combine the advantages of exponential approximation to distribution to reduce the computation time and the flexibility and efficiency of MBL to model high-dimensional and sparse count data.

5.3.2 The Exponential Multinomial Beta-Liouville (EMBL)

In this section, we derive a family of distributions that is an exponential family and is also an approximation to the MBL family, and we call it EMBL distributions. Given the sparsity nature of datasets represented using bag-of-words or bag-of-visual-words, it should be possible to evaluate

the probability as a function of non-zero x_d values only for computational efficiency. That is, the value of $\Gamma(x_d+1)=1$ when $x_d=0$. Moreover, we found empirically that $\alpha \ll 1$ and $\beta \simeq 1$ based on fitting MBL on different tested datasets. Thus, we can use the following fact by [12] for a small values for the parameters α, such that:

$$\lim_{\alpha \to 0} \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)} - \alpha\Gamma(\beta) = 0, \tag{5.2}$$

We can thus replace $\Gamma(\alpha+\beta)$ by $\alpha\Gamma(\beta)\Gamma(\alpha)$ in Eq.(5.1), we can write the MBL density function as:

$$\mathcal{MBL}(\mathbf{X}|\xi) \approx \frac{\Gamma((\sum_{d=1}^{D+1} x_d)+1)}{\prod_{d:x_d \geq 1} x_d!} \times \frac{\Gamma(s)\Gamma(\alpha')\Gamma(\beta')\alpha}{\Gamma(s+n)\Gamma(\alpha'+\beta')} \prod_{d:x_d \geq 1} \frac{\Gamma(\alpha'_d)}{\Gamma(\alpha_d)} \tag{5.3}$$

where $n = \sum_{d=1}^{D+1} x_d$ and $s = \sum_{d=1}^{D} \alpha_d$.

Recalling that $\alpha'_d = \alpha_d + x_d$, and considering that α_d is actually very small in case of high dimensions [64], we can use the previous fact in Eq.(5.2) as a highly accurate approximation to replace $\Gamma(\alpha_d + x_d)/\Gamma(\alpha_d)$ by $\alpha_d\Gamma(x_d)$. Now, we can obtain the EMBL distribution using the fact that $\Gamma(x) = (x-1)!$, as:

$$\mathcal{EMBL}(\mathbf{X}|\xi) = n! \frac{\Gamma(s)\Gamma(\alpha')\Gamma(\beta')\alpha}{\Gamma(s+n)\Gamma(\alpha'+\beta')} \prod_{d:x_d \geq 1} \frac{\alpha_d}{x_d} \tag{5.4}$$

In practice the probabilities given by Eq.(5.4) are very close to those given by Eq.(5.1). Indeed, EMBL is a good approximation for real count data because of it sparsity nature or small counts for most words, as well as, it shows consistent burstiness behavior. We can notice from Eq.(5.4) that a probability of a document, or an image, is proportional to $\prod_{d:x_d \geq 1} \alpha_d/x_d$. Hence, EMBL shares the same insights gains from the EDCM, which also can be carried over to the MBL, including that the first appearance of a word d reduces the probability of a document by α_d and that this form distinguishes between word types and word tokens; for details see ([12], Section 3).

A member of an exponential family of distributions for random variables $\mathbf{X}$ indexed by a parameters set ξ, can be written as:

$$P(\mathbf{X}|\theta) = h(x)g(\xi)\exp\{\Phi(\xi)f(x)\} \tag{5.5}$$

where $\Phi(\xi)$ is called the natural parameter, $f(x)$ is the sufficient statistic, $h(x)$ is the underlying measure and $g(\xi)$ is called log normalizer which ensures that the distribution integrates to one [14]. We can write EMBL in this form as:

$$\begin{aligned}\mathcal{EMBL} \propto & \left(\prod_{d:x_d\geq 1} x_d^{-1}\right) n!\frac{\Gamma(s)\Gamma(\alpha')\Gamma(\beta')\alpha}{\Gamma(s+n)\Gamma(\alpha'+\beta')} \\ & \times \exp\left[\sum_{d=1}^{D} I(x_d \geq 1)\log(\alpha_d)\right]\end{aligned} \tag{5.6}$$

where $I(x_d \geq 1)$, the sufficient statistic, is an indicator that represents whether the word d appears at least once in the vector $\mathbf{X}$. Having the distribution in exponential form provides a numerous of the desirable statistical and computational properties of exponential family of distributions including the sufficiency that retains the essential information in a dataset regarding the parameters which reduce the complexity and computational efforts especially for sparse high-dimensional data.

5.3.3 The Learning Approach for EMBL Mixture Model

5.3.3.1 Estimating the Number of Components

For real-world clustering problems, it is important to estimate the true number of mixture components to achieve a superior performance. Formally, let $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ be a set of data controlled by a mixture of EMBL distributions with parameters ξ, where each observation is an independent count vector, the complete data likelihood corresponding to a K-component EMBL mixture is given by:

$$Q(\mathcal{X}|\boldsymbol{\xi}) = \prod_{i=1}^{N}\Big(\sum_{j=1}^{K}\pi_j\,\mathcal{EMBL}(\mathbf{X}_i|\boldsymbol{\xi}_j)\Big) \tag{5.7}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ represents the vector of mixing probabilities which are positive and sum to one. According to information theory, the optimal number of clusters K is the candidate value which minimizes the amount of information, measured in $nats$ using the natural logarithm, to transmit $\mathcal{X}$ efficiently from a sender to a receiver [71, 188]. The problem of estimating the parameters can be formulated into a transmission encoding problem. Thus, the criterion is to minimize the two-part message to be transmitted whose length is given by:

$$\text{length}(\mathcal{X}, \xi) = \text{length}(\xi) + \text{length}(\mathcal{X}|\xi) \tag{5.8}$$

Before the transmission, the observations and the parameters have to be quantized to finite precision. This quantization sets a trade-off between the two terms of the previous equation, which corresponds to the minimum message length criterion. The formula for the message length for a mixture of distributions, with N_p free parameters, is given by [40, 41]:

$$\xi_{MML} = \underset{\xi}{\text{argmin}} \left\{ -\log(P(\xi)) - \log(P(\mathcal{X}|\xi)) + \frac{1}{2}\log|F(\xi)| + \frac{N_p}{2}\left(1 + \log\frac{1}{12}\right) \right\} \tag{5.9}$$

where $F(\xi)$ is the expected Fisher Information Matrix (FIM), and $P(\mathcal{X}|\xi)$ is the complete data likelihood. In the case of EMBL mixture, the complete FIM has a block-diagonal structure as:

$$F(\xi) = N \text{ block-diag}\{\pi_1 F(\xi_1), \ldots, \pi_K F(\xi_K), M\} \tag{5.10}$$

where $F(\xi_j)$ is the Fisher matrix for a single observation known to have been produced by the jth component, and M is the Fisher matrix of a multinomial distribution which is determinant is given by $|M| = (\pi_1 \pi_2 \ldots \pi_K)^{-1}$ [179]. In case of mixture models, we make a general assumption that the parameters of the different components as a prior are independent from the mixing probabilities, and the components of $P(\xi_j)$ are independent as well [73], that is:

$$P(\xi) = P(\pi_1, \ldots, \pi_K) \prod_{j=1}^{K} P(\xi_j)$$

Giving the lack of knowledge about mixture parameters, we adopt the standard non-informative

Jeffreys' prior [189], as:

$$P(\xi_j) \propto \sqrt{|F(\xi_j)|} \tag{5.11}$$

$$P(\pi_1, \ldots, \pi_K) \propto \sqrt{|M|} = (\pi_1 \pi_2 \ldots \pi_K)^{-1/2} \tag{5.12}$$

With choosing these prior distributions and noticing that for a K-component EMBL, the number of free parameters $N_p = K(C) + K$ and $C = D + 2$ is the number of parameters specifying each component, we obtain the following optimization problem:

$$\xi_{MML} = \underset{\xi}{\operatorname{argmin}} \left\{ \frac{C}{2} \sum_{j=1}^{K} \log \pi_j - \log(Q(\mathcal{X}|\boldsymbol{\xi})) + \frac{K(C)+K}{2} \left(1 + \log \frac{N}{12}\right) \right\} \tag{5.13}$$

5.3.3.2 The Component-wise Expectation Maximization (CEM)

Starting with a large value of K may lead to several empty components and there will be no need to estimate, and transmit, their parameters. Thus, we adopt the component-wise EM procedure (CEM) [187], as proposed in [1], where we run both E and M steps for one component before moving to the next one. We may notice that Eq.(5.13) does not make sense if any of the π_j is allowed to be null. That is, we first evaluate the posterior probability $\hat{z}_{ij}$ for each competent in the E-step, as in the standard EM, according to a Bayes law:

$$\hat{z}_{ij}^{(t)} = \frac{P(\mathbf{X}_i|\xi_j^{(t)})\ \pi_j^{(t)}}{\sum_{j=1}^{K} P(\mathbf{X}_i|\xi_j^{(t)})\ \pi_j^{(t)}}, \tag{5.14}$$

Then, we need to estimate the mixture proportion for that component as:

$$\hat{\pi_j}^{(t+1)} = \frac{\max\left\{0, \left(\sum_{i=1}^{N} \hat{z}_{ij}^{(t)}\right) - \frac{C}{2}\right\}}{\sum_{j=1}^{K} \max\left\{0, \left(\sum_{i=1}^{N} \hat{z}_{ij}^{(t)}\right) - \frac{C}{2}\right\}}, \tag{5.15}$$

Any weak component, not supported by the data, with $\hat{\pi_j}^{(t+1)} = 0$ will be annihilated and does not contribute to the log-likelihood, thus, their parameters become irrelevant. Immediately, the annihilated component probability mass is redistributed to the other components increases their

chance to survive.

Let $\mathcal{K}^+$ denotes the number of non-zero components. When one component $j \in \mathcal{K}^+$, namely those for which $\hat{\pi}_j{}^{(t+1)} > 0$, their parameters updates should be performed by maximizing the log-likelihood of expected complete-data. For, α_{jd}, the updates are closed-form expression obtained by setting the partial derivative of the log-likelihood to zero and solving for α_{jd} which gives:

$$\alpha_{jd}^{(t+1)} = \frac{\sum_{i=1}^{N} \hat{z}_{ij}^{(t)} I(x_{id} \geq 1)}{\sum_{i=1}^{N} \hat{z}_{ij}^{(t)} \Big(\Psi(s_j + n_i) - \Psi(s_j) \Big)} \tag{5.16}$$

Then, we can compute $s_j = \sum_d \alpha_{jd}$ by summing each side of Eq.(5.16) over all words, giving:

$$s_j^{(t+1)} = \frac{\sum_{d=1}^{D} \sum_{i=1}^{N} \hat{z}_{ij}^{(t)} I(x_{id} \geq 1)}{\sum_{i=1}^{N} \hat{z}_{ij}^{(t)} \Big(\Psi(s_j + n_i) - \Psi(s_j) \Big)} \tag{5.17}$$

The numerator in this case is the number of times a word d appears at least once in any vector of the dataset. Note that this equation can be solved numerically efficiently as it involves only a single unknown s_j. Having s_j in hand, Eq.(5.16) can be used directly to compute each individual α_{jd}.

We cannot solve the M-step for the α_j and β_j parameters analytically (*i.e.*, a closed-form solution does not exist). Thus, we have to update these parameters in accordance with the Newton Raphson method:

$$\xi_j^{(t+1)} = \xi_j^{(t)} - H_{(\xi_j)}^{-1} G \tag{5.18}$$

where H is the Hessian matrix associated with the complete data log-likelihood which needs to be transformed to its inverse, and G is the gradient vector associated with the first order derivatives (see Appendix 1).

5.3.3.3 The Complete Algorithm for EMBL Mixture Model Learning

The proposed unsupervised learning approach is outlined in Algorithm 5. In practice, we initialize the $\pi_j^{(0)}$ parameter using the K-means algorithm, and the model parameters $\alpha_j^{(0)}, \beta_j^{(0)}, \alpha_{jd}^{(0)}$ were initialized randomly. Parameters will be then updated during the CEM iterations to take their

natural values in relation to the observed data. Furthermore, the upper and lower number of components are provided, and in our experiments, we have set $K_{min} = 2$, and $K_{max} = 100$. The algorithm will rerun until $\mathcal{K}^+ \geq K_{min}$, where each iteration will run the component-wise EM until convergence. When the irrelevant components, with $\hat{\pi_j}^{(t+1)} = 0$ annihilated, the parameters are updated accordingly, and the MML criterion is re-evaluated for non-zero components only. Since each update to the parameters resulting from the E step followed by the M step is guaranteed to increase the log-likelihood function, which is equivalent to minimizing the length of the two-part message, the algorithm is deemed to have converged when the change in the message length, or alternatively in the log-likelihood function, becomes insignificant.

Algorithm 5: The complete algorithm for EMBL mixture learning with model selection.

Output: The optimal number of components K^*, best mixture model parameters $\boldsymbol{\xi}_{best}$
Input: $\mathcal{X} = \{X_1, \ldots, X_N\}$, K_{min}, K_{max}, $\boldsymbol{\xi}^{(0)} = \{\xi_1, \ldots, \xi_{K_{max}}\}$ where $\boldsymbol{\xi}_j = \{\pi_j^{(0)}, \alpha_j^{(0)}, \beta_j^{(0)}, \alpha_{jd}^{(0)}\}_{d=1}^{D}$

Set: $t \leftarrow 0$, $\mathcal{K}^+ = K_{max}$, $LEN_{min} = +\infty$;
while *Convergance criteria is no reached* **do**
  while $\mathcal{K}^+ \geq K_{min}$ **do**
    for $j = 1$ *to* $\mathcal{K}^+$ **do**
      for $i = 1$ *to* N **do**
        Compute the posterior probabilities $\hat{z}_{ij}^{(t+1)} = p(j|\mathbf{X}_i, \xi_i)$ using Eq.(5.14);
      end
      Update the mixing proportion $\hat{\pi_j}^{(t+1)}$ using Eq.(5.15);
      if $\hat{\pi_j}^{(t+1)} > 0$ **then**
        Evaluate $\xi_j^{(t+1)}$: $\alpha_{jd}^{(t+1)}$, $s_j^{(t+1)}$ using Eqs. (5.16, 5.17), and $\alpha_j^{(t+1)}$, $\beta_j^{(t+1)}$ using Eq. (5.18);
      else
        $\mathcal{K}^+ = \mathcal{K}^+ - 1$;
      end
    end
    Compute optimal length for the non-zero components $LEN_{MML}^{(t+1)}$ using Eq.(5.13) ;
    if $LEN_{MML}^{(t+1)} < LEN_{min}$ **then**
      Set $LEN_{min} = LEN_{MML}^{(t+1)}$;
      Set $\boldsymbol{\xi}_{best} = \xi^{(t+1)}$;
    end
    Set $t \leftarrow t + 1$;
  end
end

5.3.4 Perspectives on the Proposed Model Efficiency

Generally speaking, a sufficient statistic is supposed to contain by itself all of the information about the unknown parameters that the entire sample could have provided. In other words, the essential in a dataset can be characterized by the sufficiency, so we can reduce the computation time by throwing away the inessential data. The intuitive notion of sufficiency is that $f(x)$ contains all of the essential information in X regarding ξ. So only the sufficient statistic will be retained for estimating the parameters [14, 190, 191], which makes the proposed model computationally efficient, especially in high-dimensional spaces. Empirically, the proposed model has shown to be more efficient than the original MBL in terms of both computation time and memory usage. In Figure 5.1, we performed a visual complexity analysis on different datasets (described below in Section 5.4)), namely, IMDB, Swedish Leafs, High Five, and Caltech Faces. The complexity estimation, indeed, depends on the size of the dataset (*i.e.*, number of observations N), and the number of components K. Thus, the overall computation complexity for one iteration of EMBL is $O(NK)$.

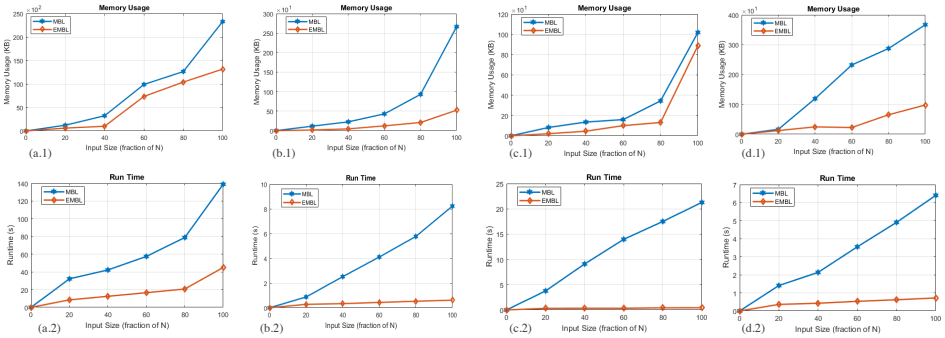


Figure 5.1: Visualizing the performance of the proposed algorithm (EMBL) and original MBL against different input sizes considering memory usage (top row), and run time (bottom row) for (a) text dataset (IMDB: $N = 25,000$ $K = 2$), (b) shape dataset (Swedish Leafs $N = 600$ $K = 15$), (c) video dataset (High Five $N = 100$ $K = 4$), and (d) image dataset (Caltech Faces $N = 200$ $K = 2$).

Furthermore, the proposed model based on MBL is an efficient generative model for count data for several reasons. First, it is a fixable alternative to the Dirichlet Compound Multinomial (DCM) [9, 63] that shares the same advantages over a generic multinomial (MN) distribution, which is typically used to model count data. More precisely, it can handle the burstiness phenomenon and the overdispersion, which MN fails to handle given its independency assumption [192]. Moreover,

like the generalized Dirichlet, it can overcome the main restrictions of the Dirichlet distribution, including the negative-correlation and the equal-confidence requirements. Finally, MBL and EMBL are attractive generative models that have fewer parameters than the other hierarchical Bayesian frameworks which use other generalization of Dirichlet as prior for the Multinomial; *i.e.,* MSD [23], and MGD [10] with comparable performance. Another important feature of the proposed algorithm is that it is less initialization dependent comparing to the standard EM. Given that it starts with a large number of components, which is usually much larger than the optimal number, we avoid the local maxima of the likelihood that arises when there are too many components in one region of the space and too few in another [1, 183]. The second point of view is that the component annihilation in M-step (Eq. 5.15) makes the algorithm automatically avoids the possibility of convergence toward a singular estimate at the parameter space boundary [1]. Furthermore, although CEM seems to be much computationally heavier than standard EM, due to the multiple E-step to recompute the posterior probabilities, it is actually not [180, 187].

On the other hand, the proposed algorithm has some limitations as follows. First, we consider the maximum likelihood estimation approach for learning the parameters, where the learning process can be extended to Bayesian or variational estimation, which computes (an approximation to) the entire posterior distribution of the parameters and latent variables. Second, we assumed that the data is static, while in many real-world applications, data are received in online mode; thus, the online clustering approach supports life-long learning. In addition, our model is based on clustering the data into a finite number of components, while it would be better to make it infinite (*i.e.*, to let the number of mixture components increases as new vectors arrive). Finally, we considered that all non-zero features have the same weight, but in practice, some features are not contributing to (or even degrading) the clustering process. Thus, future work may consider a feature selection approach to select the best features subset for improving the performance further.

5.4 Experimental Results

Information explosion is not only creating massive amounts of data but also a diverse format of data. For instance, social media platforms offer many possibilities of data formats, including

textual data, pictures, videos, sounds, and geolocations. Analyzing different types of data can help in gaining insights into issues, trends, influential actors, and other kinds of information. Thus, in our experiments, we have considered different datasets to prove the merits of the work and its usefulness in real-world applications. The first application concerns text classification, in particular, sentiment analysis. The second one involves shape clustering using the shape context descriptor. In the third application, we focus on the problem of recognizing human interaction in realistic videos from movies and TV shows. Finally, we address the problem of distinguishing genders from human faces, by developing flexible probabilistic SVMs kernels based on the proposed mixture of EMBLs. All the experiments were conducted using optimized MATLAB R2017a codes on an Intel(R) Core(TM) i7-6700 Processor PC with the Windows 7 Enterprise Service Pack 1 operating system with a 16 GB main memory. The results that we will present in the following represent the average over 20 runs of the different learning algorithms.

5.4.1 Sentiment Analysis

With the explosive growth of social media (*e.g.*, reviews, forum discussions, blogs, micro-blogs, Twitter, comments, and postings in social network sites) on the Web, individual customers and organizations increasingly rely on the content in these media for decision making. Sentiment analysis, also called opinion mining, involves analyzing evaluations, attitudes, and emotions, expressed in a piece of text, towards entities such as products, services, or movies [193]. We investigate sentiment analysis at the document level, such that our task is to classify whether a whole opinion document expresses a positive or negative sentiment. The challenges in sentiment analysis, as a text clustering application, include that the reviews are usually limited in length, have many misspellings, and shortened forms of words. Thus, the vocabulary size is very large, and the count vector that represents each review will be highly sparse. Our experiments aim at comparing the proposed algorithm to other clustering methods such as the Spherical k-Means (SKM), the Gaussian mixture model (GMM), and the mixture of Multinomials (MM) to give a baseline of the difficulty of the problem. Moreover, we compare to other hierarchical Bayesian modeling frameworks including mixtures of DCM [9], EDCM [12], MGD [10], MBL [11], and MSD [23], that have been previously proposed

Table 5.1: Clustering results for the IMDB dataset using EMBL mixture.

Model	Precision	Recall	F-score	Mutual info.	time
SKM	55.90±0.03	55.91±0.07	55.90±0.02	0.5800±0.03	252.76
GMM	61.40±0.05	61.40±0.04	61.40±0.03	0.6719±0.07	169.48
MM	64.18±0.05	64.40±0.06	64.29±0.02	0.6520±0.03	143.52
DCM	71.14±0.05	89.45±0.05	79.25±0.02	0.8578±0.09	227.11
MSD	76.44±0.02	84.54±0.04	80.29±0.03	0.8432±0.03	254.46
MGD	75.55±0.02	81.43±0.07	78.38±0.02	0.8992±0.05	338.26
MBL	83.69±0.02	83.99±0.03	83.84±0.02	0.8927±0.06	139.91
EDCM	78.54±0.09	89.33±0.14	83.59±0.04	0.8861±0.07	32.07
EMBL	83.75±0.02	84.60±0.02	84.17±0.05	0.8940±0.06	45.60

for modeling count data. The performances of the different mixture models are compared according to the macro averaged recall, precision, and F-score values, whose definitions can be found, for instance, in [194], as well as the mutual information [65]. Moreover, we reported the time (in seconds) for a single run to the convergence of an optimized MATLAB code.

We have used different large scale datasets that have been recently constructed, namely; IMDB movie reviews [115], Amazon, and Yelp reviews [195]. Ratings on IMDB are given as star values $\in \{1, 2, \ldots, 10\}$, which were linearly mapped to $[0, 1]$ to use as document labels; negative and positive, respectively. We used a union of the training and testing sets having around $25,000$ samples from each positive/negative group with $76,340$ unique words in total. The Amazon reviews dataset consists of reviews that span a period of 18 years, and it includes product and user information, ratings, and a plain text review. The Amazon reviews full score dataset is constructed by randomly taking $600,000$ for training samples and $130,000$ testing samples for each review score from 1 to 5. We used a union of the training and testing sets having a total of $50,000$ samples from all the five groups with a vocabulary size of $55,383$ unique words. The Yelp dataset contains a polarity label by considering stars 1 and 2 negative, and 3 and 4 positive reviews about local businesses. The full dataset has $280,000$ for training samples and $19,000$ test samples in each polarity. We considered a subset with a total of $20,000$ sentiments randomly, and equally, selected from each polarity with $85,638$ unique words.

The clustering results for the three datasets are given in Tables (5.1-5.3). According to the F

Table 5.2: Clustering results for the Amazon dataset using EMBL mixture.

Model	Precision	Recall	F-score	Mutual info.	time
SKM	62.53±0.02	65.23±0.03	63.85±0.02	0.7280±0.02	138.70
GMM	64.65±0.02	74.04±0.02	69.03±0.03	0.5366±0.03	122.43
MM	50.83±0.02	51.99±0.04	51.91±0.02	0.6847±0.05	94.91
DCM	55.65±0.05	63.94±0.02	59.51±0.02	0.8045±0.02	180.42
MSD	83.56±0.02	83.57±0.03	83.57±0.02	0.8468±0.04	148.26
MGD	82.52±0.03	82.53±0.01	82.53±0.01	0.8241±0.02	240.86
MBL	82.37±0.02	82.78±0.02	82.57±0.02	0.8303±0.03	101.46
EDCM	80.65±0.03	80.88±0.01	80.77±0.03	0.8114±0.05	30.50
EMBL	82.20±0.03	82.42±0.02	82.31±0.03	0.8295±0.01	35.17

Table 5.3: Clustering results for the Yelp dataset using EMBL mixture.

Model	Precision	Recall	F-score	Mutual info.	time
SKM	74.08±0.02	74.08±0.02	74.08±0.02	0.6209±0.04	63.45
GMM	63.21±0.02	77.00±0.02	69.42±0.02	0.7271±0.04	60.87
MM	89.12±0.01	89.20±0.02	89.16±0.02	0.8527±0.03	18.45
DCM	91.01±0.03	91.01±0.03	91.01±0.03	0.8311±0.02	58.57
MSD	91.01±0.03	91.01±0.03	91.01±0.03	0.8911±0.02	36.13
MGD	91.00±0.07	91.01±0.04	91.00±0.03	0.8909±0.03	123.15
MBL	90.47±0.04	90.66±0.01	90.57±0.01	0.8978±0.02	50.65
EDCM	89.25±0.02	89.28±0.02	89.27±0.02	0.8328±0.04	10.86
EMBL	94.05±0.02	94.41±0.01	94.72±0.02	0.8945±0.02	14.00

measures in these tables, we can see that the mixture of EMBL behaves similarly to MBL, MGD, and MSD, which themselves outperform the other compared models. However, it has shown to be much faster (*i.e.*, on average, the proposed algorithm is 3-8 times faster than the other models with comparable performance). Although the EDCM is computationally efficient, the mixture of EMBL outperforms EDCM, as shown by the F measures and mutual information in the tables. Moreover, a comparative study between the proposed framework and other approaches for text clustering from the state-of-the-art is depicted in Table 5.4. These successful approaches include the character level Convolutional model (CNN-char) [195], long short-term memory with Gated Recurrent Neural Network (LSTM-GRNN) [196], the very deep convolutional network (VDCNN)

Table 5.4: Comparison of our method to the best published results (avg accuracy %) from previous works for sentiment analysis.

	Datasets		
	IMDB	Amazon	Yelp
CNN-char [195]	-	59.6	62.00
SVM + Bigrams [196]	40.90	-	62.40
LSTM-GRNN [196]	45.30	-	67.60
Fast text + bigrams [197]	-	60.20	95.70
VDCNN [198]	-	63.00	95.70
HN-ATT [199]	49.40	63.60	71.00
Mixture of exponential MBL	84.14	82.31	94.72

of [198], and the hierarchical attention networks (HN-ATT) by [199]. According to the reported results, the proposed framework gives a superior performance.

5.4.2 Shape Clustering Using Shape Context

In today's world, large amounts of digital images and videos are increasingly generated, which induce an urgent need for the development of automatic methods to analyze and index these overwhelmingly digital datasets. The shape is well-known to be a strong discriminating feature; thus, it is an important cue for several computer vision applications such as object recognition, matching, content-based image retrieval, and indexing (see, for instance, [200–202]). A great deal of material dealing with shape modeling has been published, and several shape descriptors have been proposed in the past, yet they can be grouped into three main categories [203]: contour-based descriptors, image-based descriptors, and skeleton-based descriptors. Contour-based descriptors are based on the mapping of the contour of a given object to some representation from which a shape descriptor is derived.

In our experiments, we have used an interesting descriptor, called shape context, which has been proposed by [204]. In this approach, an object is assumed to be essentially captured by a finite set of its points N sampled from the internal or external contours on the object. These points are considered as locations of edge pixels as found by an edge detector. To detect the edges, we first applied a Gaussian filter to smooth the image in order to remove the noise and find the intensity gradients of the image. Then, we applied non-maximum suppression to thin the resulting edges

Table 5.5: Shape clustering performance (avg. accuracy %) using different generative models.

Dataset	MM	DCM	MSD	MGD	MBL	EDCM	EMBL
MPEG	75.90	76.92	77.89	77.89	78.05	83.86	85.50
Leafs	87.99	88.87	89.40	89.05	89.02	94.45	96.50

and discard any weak edges according to a specific threshold. The shape context is then obtained as a vector of the relative positions between each point and the other $N-1$ points. As choosing more points will result in an accurate representation of the shape, we sampled 200 points from the internal and external boundary of each shape image, and for each point's shape context, we used five bins for the log-distance and 12 bins for relative orientation, which leads to 60 bit vectors for each point. Then, as done in [205], we considered each context vector as a visual word and created the Bag-of-Features (BoF) [89].

In order to illustrate the effectiveness and efficiency of the proposed model in clustering shapes, experiments were conducted on different datasets include: **MPEG7CE-1** Set B shapes dataset [206], and **Swedish Leaf** dataset [207]. Samples from both datasets are shown in Fig. 5.2. MPEG7 CE-1 Set B consists of 1400 shapes representing 70 real-life objects, with 20 similar shapes for each class. The challenge is this dataset that includes rotation, scaling, skew, stretching, defection, indentation, and articulation of shape. The Swedish leaf dataset is challenging because of its high inter-species similarity. It consists of 1125 different species of leaves in 15 categories with minimal contour-based differences. The majority of the published papers in shape categorization focused on proposing and evaluating different descriptors. For a fair comparison, we tested different generative models using the same data representation approach, as described above. A summary of the clustering accuracy is presented in Table 5.5. We note that our method performs comparably to the best generative model with a similar approach (*i.e.,* EDCM). It is interesting that EMBL is significantly better than the corresponding model (MBL) and other models that behave similarly in shape discrimination.

The results from the literature on the Swedish leaf database were summarized in Table 5.6. As shown in Table 5.6, the accuracy of recognizing the leaf type using our approach is higher than SSLDP [208], and MSDM [209], and almost comparable to the other methods, with an accuracy of 96.50%, which is a promising result considering that our approach is completely unsupervised.

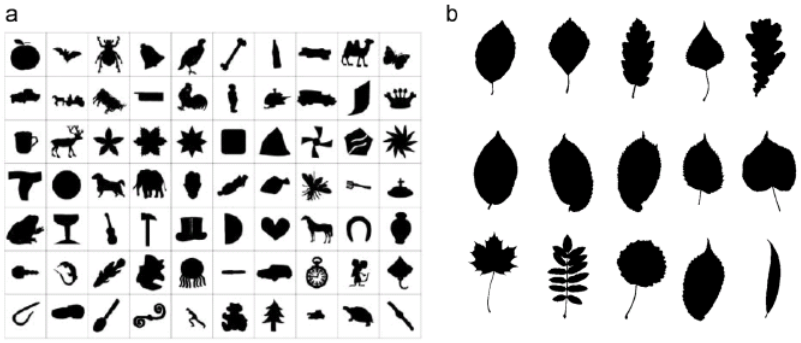


Figure 5.2: (a) MPEG7CE-1 Set B dataset representative shapes,(b) Leaf dataset representative shapes.

Table 5.6: Comparison of our method with the state of the art for the Swedish leaf database.

Methods	Accuracy
SSODP [210]	97.14 %
SSLDP [208]	95.82 %
MSDM [209]	93.60%
I-IDSC [211]	97.07%
MARCH [212]	97.33%
Mixture of exponential MBL	96.50%

5.4.3 Recognition of Human Interactions in Films and TV Shows

Human activity and action recognition is a popular topic in computer vision. Previous works have focused on recognizing individual actions such as running, walking, etc. The objective of this application is to show the efficiency of our model in recognizing natural human actions in diverse and realistic video settings. In particular, we address the problem of recognizing interactions between two people in realistic scenarios, which is useful in video retrieval tasks. We have used two challenging datasets from feature films and TV shows with different human interactions. Each video in each dataset is represented as a vector of count data using the extension of the Bag of words paradigm to videos. In particular, we start by detecting the Spatio Temporal Interest Points (STIP), where the local neighborhood has significant variations in both spatial and temporal domains [213]. Then, we used 3D SIFT descriptor [214] that has shown to accurately captures the Spatio-temporal nature of the video data. Each dataset is split evenly into two groups, for constructing the codebook and representation.

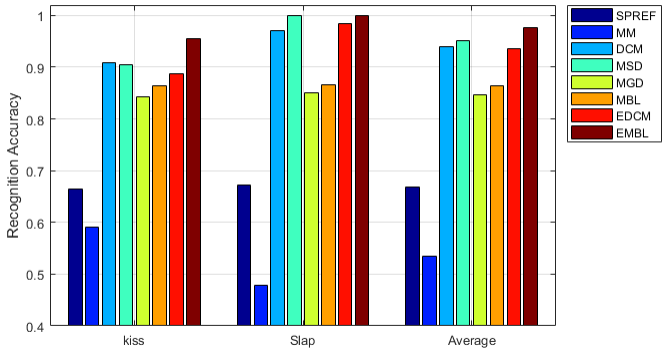


Figure 5.3: Average and Intra-class accuracy for both actions in Kiss and Slap dataset.

The **Kiss and Slap** dataset [215] consists of actions performed in a range of film genres consisting of classic old movies such as "A Philadelphia Story","The Three Stooges", and "Gone With the Wind", comedies such as "Meet the Parents", and romantic films such as "Connie and Carla". This dataset provided a representative pool of natural samples of action classes such as Kissing (92 samples), and Hitting/slapping (112 samples) appeared in a wide range of scenes and viewpoints and were performed by different actors. We compared our proposed approach and different generative models to a previous work using Spatio Temporal Regularity Flow (SPREF) [215]. Figure 5.3 shows the average percentage of correctly classified clips for each interaction class, where we can see clearly that the proposed model provides the highest average accuracy for both classes and notably outperforms the other models. Moreover, the average run time for recognizing human interaction using the proposed exponential MBL is 0.737 seconds, which is around 5 times faster than the corresponding MBL model that takes 3.832 seconds for the same task.

TV Human Interactions Dataset [216] was created by the Visual Geometry Group in 2010 from over 20 different TV shows. In the context of human interaction recognition, several challenges must be addressed, including, for example, the background clutter, the varying number of people in the scene, camera motion, and changes of camera viewpoints. Two hundred of the clips contain one of four interactions: handshake (HS), high five (HF), hug (HG), and kiss (KS), each appearing in 50 videos (snapshots samples of each interaction are shown in 5.4). Each video clip length ranges from 30 to 600 frames. A summary of the average precision for learning the human interaction in the High Five dataset, as well as the average run-time, are shown in Table 5.7. Significantly, the

Table 5.7: Average precision results and time for recognizing human interaction in High Five dataset using different generative models .

Model	HS	HF	HG	KS	AVG	Time (s)
MM	0.2000	0.4000	0.0400	0.4800	0.2800	0.160
DCM	0.4903	0.7941	0.4758	0.4398	0.5500	20.84
MSD	1.0000	1.0000	0.9600	0.8000	0.9400	23.00
MGD	0.6814	0.6635	0.6551	0.6000	0.6500	20.47
MBL	0.9200	0.7600	0.7200	1.0000	0.8500	20.97
EDCM	1.0000	0.9600	1.0000	0.8400	0.9500	0.449
EMBL	1.0000	1.0000	1.0000	0.9600	0.9900	0.437

mixture of EMBL outperforms the similar approach of EDCM, which itself outperforms the other models in recognizing the human interaction in all the classes.

Figure 5.4: High Five dataset snapshots with different scale and camera views.

Table 5.8 compares our method with state of the art. High Five typically serves as a good testbed for various structure model applied for action recognition. The previously published results are around 50-60%. With our framework, we achieve 99.00% on this challenging dataset.

Table 5.8: Comparison of our method with the state of the art for High Five dataset.

	Average Precision
SVM [217]	27.48%
BoF+Structured SVM [217]	54.76%
Propagate Hough Voting [218]	56.00%
Spectral Divisive K-Means [219]	64.00%
Feature Encoding [220]	69.40%
Space-Time Tree Ensemble [221]	64.40%
Mixture of exponential MBL	99.00%

5.4.4 Distinguishing Male and Female Faces Using Generative Kernels

5.4.4.1 Learning Approach and Datasets

Considering the capabilities and limitations of both generative and discriminative approaches, there have been hybrid methods to combine the advantages and desirable properties of both [131, 141]. While the generative models (*e.g.,* mixture models and hidden Markov models) aim to estimate the class-conditional distributions, the discriminative approaches focus directly on the classification problem by estimating a classification function. Given their good discrimination and generalization capabilities, Support Vector Machines (SVMs) are well known powerful tools for pattern classification. Indeed, the performance of an SVM largely depends on the kernel function it adopts. Thus, an important issue in applying this classifier is the choice of the kernel function, $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ for non-separable data [222]. The idea is to capture the intrinsic properties of the data to classify based on a similarity measure between input vectors taking into account a prior knowledge of the problem domain. In this section, we develop kernels based on EMBL mixture models, and could also be called generative kernels [163], that address some practical limitations of classical kernels (*e.g.,* linear, radial basis function, and polynomial). We show the capability of the generated kernel function in the problem of recognizing human faces that require handling bags of count vectors.

We used three standard and challenging face recognition databases, as follows. The first is **AR** face dataset [223], which has 4,000 color images corresponding to 126 people's faces (70 men and 56 women). Images feature frontal view faces with different facial expressions, illumination conditions, and occlusions (sunglasses and scarf). Second dataset is the one created by **AT&T** laboratories Cambridge [224]. It contains a set of an upright, frontal position face images in grayscale with ten different images of each of 40 distinct subjects, *e.g.,* open / closed eyes, smiling / not smiling, and glasses / no glasses. The last dataset is **Caltech** faces by California Institute of Technology[1], consists of 450 face images of around 27 unique people (both genders) with different lighting/expressions/backgrounds. In the following, we present the different kernels generated from our proposed model.

[1] http://www.vision.caltech.edu/html-files/archive.html

AR database Caltech database

Figure 5.5: Samples from the face recognition datasets.

Fisher Kernels The Fisher kernel is mainly based on exploiting the geometric structure on the statistical manifold by mapping each individual sequence into a single feature vector, defined in the gradient log-likelihood space, as initially proposed in [141]. Let $\mathcal{O} = \{\mathbf{O_1}, \ldots, \mathbf{O_N}\}$ to be a set of multimedia objects (*e.g.,* images), where each image $\mathbf{O}_n$ is defined by a sequence of feature vectors of count data $\mathcal{X}_{\mathbf{O_n}} = \{X_{O_n 1}, \ldots, X_{O_n T}\}$. Each individual object $\mathcal{X}_{\mathbf{O_n}}$ has its own size T as the image can be represented by a bag of pixel vectors of a set of local descriptors [164, 165]. The resulted feature vector is called the Fisher score and defined as: $U_{\mathcal{X}_{\mathbf{O_n}}} = \frac{\partial p(\mathcal{X}_{\mathbf{O_n}}|\Theta)}{\partial \Theta}$, where each component is the derivative of the log-likelihood with respect to a particular parameter. In the case of finite mixture model of EMBLs; the corresponding feature space is $(K(D+3)-1)$-dimensional. The kernel is then defined as: $\mathcal{K}(\mathcal{X} : \mathcal{X}_{O_n}) = U_{\mathcal{X}} F(\Theta)^{-1} U_{\mathcal{X}_{O_n}}$, where $F(\Theta)$ is the Fisher information matrix whose role is less significant and then can be approximated by the identity matrix [141]. The gradient of $\log P((\mathcal{X}|\Theta)$ with respect to the EMBL model parameters, is calculated by straightforward manipulations as shown in Eq.(5.16), (5.27), and (5.28). Furthermore, computing the gradient $\pi_j, j = 1, \ldots, K$, which is the same for any mixture model, gives:

$$\frac{\partial \log(\mathcal{X}|\Theta)}{\partial \pi_j} = \sum_{t=1}^{T} \left[\frac{z_{tj}}{\pi_j} - \frac{z_{tj}}{\pi_1} \right], \quad j = 2, \ldots, K \tag{5.19}$$

Considering the unity constraint on mixing weights, we have only $K-1$ free parameters, which explains the fact that the previous gradient equation is defined for $k \geq 2$ as π_1 can be determined knowing the values of the other mixing parameters ($\pi_1 = 1 - \sum_{j=2}^{K} \pi_j$).

Bhattacharyya Kernel We generate a probability product kernel where the kernel in the original sequence is replaced by computing the probability density functions (PDFs) space [149, 166]. Let $\mathcal{X}$, and $\mathcal{X}'$ be two sequences of feature vectors representing two multimedia objects defined on the space Ω (where Ω is the D-dimensional simplex in the case of EMBL distribution). That is, the kernel becomes a measure of similarity between probability distributions as the following : $\mathcal{K}(\mathcal{X}, \mathcal{X}') \Rightarrow \mathcal{K}_\rho(p(X), p'(X')) = \int_\rho p(X)^\rho, p'(X')^\rho dX$, where ρ is a parameter. An important special case of probability product kernels (when $\rho = 1/2$) is the Bhattacharyya kernel, originally proposed by [225] which, despite its cubic complexity, has the main advantage of nonlinear flexibility [166]. The Bhattacharyya kernel is defined as follows:

$$\mathcal{K}_{BH}\Big(p(\mathbf{X}|\theta_j), p(\mathbf{X}|\theta_l)\Big) = \int_0^{+\infty} \sqrt{p(\mathbf{X}|\theta_j), p(\mathbf{X}|\theta_l)} dX \tag{5.20}$$

Given the fact the EMBL belongs to the exponential family of distribution, we could find a closed form for this kernel, which is given in (Appendix 2).

Information Divergence Kernels Another alternative for generative SVM kernels is the one based on the information divergence distance such as the Kullback–Leibler (KL) kernel [148]. Probabilistic kernels based on the symmetric Kullback–Leibler divergence have been successfully applied for different multimedia classification tasks based on both Gaussian and non-Gaussian mixtures [25, 144, 150]. The symmetric Kullback–Leibler divergence between $p(\mathbf{X}|\Theta_j)$ and $p(\mathbf{X}|\Theta_l)$ is given by:

$$\mathcal{K}_{KL}\Big(p(\mathbf{X}|\theta_j), p(\mathbf{X}|\theta_l)\Big) = \exp[-A\ J(p(\mathbf{X}|\theta_j), p(\mathbf{X}|\theta_l))] \tag{5.21}$$

where A is a kernel parameter included for numerical stability, and

$$J(p(\mathbf{X}|\theta_j), p(\mathbf{X}|\theta_l)) = KL(p(\mathbf{X}|\theta_j), p(\mathbf{X}|\theta_l)) + KL(p(\mathbf{X}|\theta_l), p(\mathbf{X}|\theta_j))$$

The KL divergence has a closed-form expression in the case of the EMBL distribution and is given in (Appendix 3).

Rényi and Jensen-Shannon Kernels We also derive two other special probabilistic kernels that are considered as a generalization of the symmetric Kullback-Leibler kernel, namely; the Reńyi and Jensen-Shannon kernels [149]. The Rényi kernel is based on the symmetric Reńyi divergence [153], as:

$$\mathcal{K}_R\Big(p(\mathbf{X}|\theta_j), p(\mathbf{X}|\theta_l)\Big) = \Bigg[\int_0^{+\infty} p(\mathbf{X}|\theta_j)^{\sigma} p(\mathbf{X}|\theta_l)^{1-\sigma} dX \times \int_0^{+\infty} p(\mathbf{X}|\theta_l)^{\sigma} p(\mathbf{X}|\theta_j)^{1-\sigma} dX\Bigg]^{A/(1-\sigma)} \tag{5.22}$$

where $\sigma > 0$ and $\sigma \neq 1$ is the order of Rényi divergence. In the case of an EMBL distribution, we can find a closed-form expression for the Rényi divergence, as shown in (Appendix 4).

The second kernel is the Jensen-Shannon (JS) Kernel, generated according to the Jensen-Shannon divergence [154], and is given by [149]:

$$\mathcal{K}_{JS}\Big(p(\mathbf{X}|\theta_j), p(\mathbf{X}|\theta_l)\Big) = \exp[-A\ JS_{\omega}(p(\mathbf{X}|\theta_j), p(\mathbf{X}|\theta_l))] \tag{5.23}$$

where:

$$JS_{\omega}(p(\mathbf{X}|\theta_j), p(\mathbf{X}|\theta_l)) = H\left[\omega\ p(\mathbf{X}|\theta_j) + (1-\omega) p(\mathbf{X}|\theta_l)\right] - \omega\ H\left[p(\mathbf{X}|\theta_j)\right] - (1-\omega)\ H\left[p(\mathbf{X}|\theta_l)\right] \tag{5.24}$$

where ω is a parameter, and:

$$H[p(\mathbf{X}|\theta_j)] = -\int_0^{+\infty} p(\mathbf{X}|\theta_j) \log p(\mathbf{X}|\theta_j) dX \tag{5.25}$$

is the Shannon entropy and we can show, that in the case of the EMBL distribution, a closed-form is existed (see Appendix 5).

5.4.4.2 Generative Models Validation via BOF Approach

The objective of our first set of experiments is to show the merits of our generative model using the Bag-of-Features (BoF) approach [89]. Each dataset was randomly split into two halves to construct visual vocabulary and representation. Each image is then represented by a vector describing the frequencies of a set of visual words, provided from the constructed visual vocabulary. For this objective, we compare our proposed model to different generative models. Moreover, we compare the classification results obtained by KNN, by setting the number of neighbors K to 3, 5 and 9, as well as, SVMs with different classic kernels optimized for each dataset namely polynomial kernel (SVM-p), sigmoid kernel (SVM-s), and RBF kernel (SVMr). A summary of the classification and clustering results obtained for the different tasks is shown in Table 5.9 and Figure 5.6, respectively. Table 5.9 summarizes the classification results, measured by the average values of the diagonal entries of the confusion matrices, obtained for the different classification approaches. The best results for SVM were obtained using the polynomial kernel with an average accuracy of 88.89%, 87.14%, and 87.18% for AR, AT&T, and Caltech database, respectively. Moreover, the KNN with $K = 3$ has also shown good performance on the three tested databases.

Table 5.9: Classification performance obtained for the different face datasets using different techniques considering the BOF approach.

Dataset	SVM-p	SVM-s	SVM-r	KNN$_{(K=3)}$	KNN$_{(K=5)}$	KNN$_{(K=9)}$
AR	88.89	76.09	84.44	85.15	82.74	73.67
AT&T	87.14	80.00	74.29	90.23	87.93	87.36
Caltech	87.18	76.92	82.05	83.08	80.00	74.36

A comparison of the clustering results using different generative models is shown in Figure 5.6. We can see that both EDCM and EMBL perform better than the two other generative models, which themselves perform better than SVM with classic kernels. Moreover, EMBL is able to better differentiate between male and female faces across the different tested databases. The average accuracy obtained using a mixture of EMBLs is 98.31%, 98.96%, and 90.78% for AR, AT&T, and Caltech database, respectively.

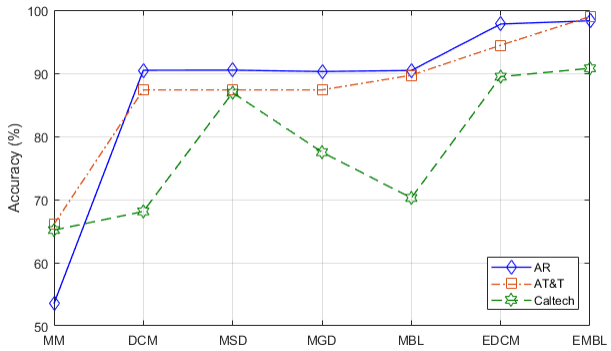


Figure 5.6: Clustering performance obtained for the different datasets using different techniques considering the BOF approach in different face recognition datasets.

5.4.4.3 Classification Results Using Generative/Discriminative Approach

In our experiments, here, we replaced the visual words generation by fitting directly our generative model, EMBL, to the local SIFT feature vectors extracted from the images (*i.e.*, each image is encoded as a bag of SIFT feature vectors). Consequently, each image was represented by a finite mixture model, which can be viewed actually as the generative stage. Then, the kernel matrices were calculated to feed the SVM classifier, which represents our discriminative stage. Moreover, the values for all design parameters were obtained by performing 5-fold cross-validation. In Table 5.10, we compare the different generated kernels based on EMBL mixture, and the previously applied kernel for discrete data generated, in the same way, from DCM mixture namely the Fisher kernel [147], and the different information divergence and fisher score kernels based on MSD/EMSD [25]. It is actually obvious from this table that our developed hybrid model is an adequate SVM kernel that is able to incorporate prior knowledge about the nature of data involved in the problem at hand and, therefore, permits good data discrimination. Results obtained when generating SVM kernels using MSD/EMSD, and MBL/EMBL mixture models were comparable and impressive if we take into account the difficulty of the problem. The best results across the different datasets obtained using the EMBL fisher kernel and the Jensen-Shannon based on EMBL. Moreover, we can notice that the Fisher kernels based on the exponential distributions (*i.e.,* EMSD, and EMBL) generally perform slightly better than the ones based on the corresponding model MSD and MBL, respectively.

In general, we can say that for all the tested datasets, we have obtained excellent and promising

Table 5.10: Performance (%) for gender faces distinguishing comparison of different generative kernels.

	Datasets		
	AR	AT&T	Caltech
DCM + Fisher Kernel	79.27	85.00	84.04
MSD + Fisher Kernel	88.46	89.88	87.94
EMSD + Fisher Kernel	95.53	96.39	89.81
EMSD + Kullback–Leibler	98.36	98.61	95.80
EMSD + Rényi Kernel	93.22	91.67	88.64
EMSD + Jensen-Shannon	99.62	97.50	92.22
MBL + Fisher Kernel	88.85	92.50	86.11
EMBL + Fisher Kernel	99.89	99.89	98.89
EMBL + Bhattacharyya Kernel	99.89	99.58	96.99
EMBL + Kullback–Leibler	99.06	98.75	97.78
EMBL + Rényi Kernel	99.81	97.50	97.44
EMBL + Jensen-Shannon	99.89	99.89	98.75

classification results as compared to the results using widely used discriminative approaches, *i.e.,* SVM with classic kernels and KNN. Moreover, our proposed hybrid frameworks are able to provide strongly acceptable results when compared to their generative counterparts, as we can see from the results in Table 5.11 which was based solely on the generative models[2].

Table 5.11: Clustering performance by fitting directly the different generative models to faces datasets.

Dataset	MM	DCM	MSD	MGD	MBL	EDCM	EMBL
AR	65.90	85.09	87.04	85.32	85.56	80.17	85.44
AT&T	69.96	77.50	89.31	89.45	89.25	85.70	85.50
Caltech	74.94	87.14	87.90	85.73	87.56	87.68	85.38

Furthermore, we compared the obtained results with other methods from the literature for AR faces dataset, in Table 5.12. According to the considered measure, *i.e.,* classification accuracy, our approach achieves competitive results to the state-of-the-art as we can notice that the proposed methods attain the highest accuracy rate.

[2]Note that the pure discriminative approach, *i.e.,* SVM with classic kernels, cannot be applied here, since each image is represented now by a set of vectors)

Table 5.12: Comparison of our method with the state of the art for the AR face dataset.

Methods	Accuracy
CRC_RLS [226]	93.70%
MRL [227]	92.83%
Progressive CNN [228]	85.62%
EMSD + Jensen-Shannon	99.62%
EMBL + Fisher, Jensen-Shannon, Bhattacharyya Kernel	99.89%
EMBL + Rényi Kernel	99.81%

5.5 Conclusion

A novel clustering framework for high-dimensional and sparse count data has been proposed in this work. The proposed model is based on an exponential-family approximation of the Multinomial Beta-Liouville distribution, which we call EMBL. The goal is to provide a more flexible framework than the previously proposed EDCM that has shown to be efficient in high-dimensional spaces. We proposed a robust learning algorithm for addressing the problems of parameters estimation and model selection simultaneously. The proposed approach successfully selected the optimal number of components, that agrees with the prespecified one, in different datasets. Experiments with different real-world applications using standard and widely used datasets have shown the effectiveness of the proposed approach.

Appendix 1: Newton Raphson Approach for EMBL

The complete data log-likelihood following EMBL mixture, is given by:

$$\begin{aligned}\log Q(\mathcal{X}|\boldsymbol{\xi}) = \sum_{i=1}^{N}\sum_{j=1}^{K}\hat{z}_{ij}\Bigg(& \log(n_i!) + \log\Gamma(s_j) + \log\Gamma(\alpha'_j) \\ & + \log\Gamma(\beta'_j) + \log(\alpha_j) - \log\Gamma(s_j+n_i) - \log\Gamma(\alpha'_j+\beta'_j) \\ & + \sum_{d=1}^{D} I(x_{id}\geq 1)\Big(\log(\alpha_{jd}) - \log(x_{id})\Big)\Bigg) \end{aligned} \tag{5.26}$$

We compute the first derivative of log likelihood for $Q(\mathcal{X}|\xi)$ with respect to α_j and β_j as:

$$\frac{\partial \log Q(\mathcal{X}|\xi)}{\partial \alpha_j} = \sum_{i=1}^{N} \hat{z}_{ij} \left(\Psi(\alpha'_j) + \frac{1}{\alpha_j} - \Psi(\alpha'_j + \beta'_j) \right) \tag{5.27}$$

$$\frac{\partial \log Q(\mathcal{X}|\xi)}{\partial \beta_j} = \sum_{i=1}^{N} \hat{z}_{ij} \left(\Psi(\beta'_j) - \Psi(\alpha'_j + \beta'_j) \right) \tag{5.28}$$

where $\Psi(.)$ is the digamma function (the logarithmic derivative of the Gamma function).

The Hessian matrix is based on the second-order derivatives calculated as follows:

$$\frac{\partial^2 \log Q(\mathcal{X}|\xi)}{\partial \alpha_j^2} = \sum_{i=1}^{N} \hat{z}_{ij} \left(\Psi'(\alpha'_j) - \frac{1}{\alpha_j^2} - \Psi'(\alpha'_j + \beta'_j) \right) \tag{5.29}$$

$$\frac{\partial^2 \log Q(\mathcal{X}|\xi)}{\partial \beta_j^2} = \sum_{i=1}^{N} \hat{z}_{ij} \left(\Psi'(\beta'_j) - \Psi'(\alpha'_j + \beta'_j) \right) \tag{5.30}$$

$$\frac{\partial^2 \log Q(\mathcal{X}|\xi)}{\partial \alpha_j \beta_j} = \sum_{i=1}^{N} \hat{z}_{ij} \left(- \Psi'(\alpha'_j + \beta'_j) \right) \tag{5.31}$$

where $\Psi'(.)$ is the trigamma function. After calculating the derivatives using Equations (5.27)-(5.31), the parameters updates will be evaluated using Newton Raphson technique.

Appendix 2: The Bahhtacharyya Kernel for EMBL

It is possible to compute the Bhattacharyya kernel in closed form for densities that belong to the exponential family of distributions, as:

$$\mathcal{K}_{BH} = \exp \left[\frac{1}{2}\Phi(\theta_j) + \frac{1}{2}\Phi(\theta_l) - \Phi\left(\frac{1}{2}\theta_j + \frac{1}{2}\theta_l \right) \right] \tag{5.32}$$

In the case of EMBL, we can show that:

$$\begin{aligned}
\mathcal{K}_{BH} = \exp\Bigg[& \frac{1}{2}\left(\log\Gamma(s_j) + \log\Gamma(\alpha'_j) + \log\Gamma(\beta'_j) + \log(\alpha_j) - \log\Gamma(s_j+n) - \log\Gamma(\alpha'_j+\beta'_j)\right) \\
& + \frac{1}{2}\left(\log\Gamma(s_l) + \log\Gamma(\alpha'_l) + \log\Gamma(\beta'_l) + \log(\alpha_l) - \log\Gamma(s_l+n) - \log\Gamma(\alpha'_l+\beta'_l)\right) \\
& - \log\Gamma\left(\sum_{d=1}^{D}\left(\frac{1}{2}\alpha_{jd} + \frac{1}{2}\alpha_{ld}\right)\right) - \log\Gamma\left(\frac{1}{2}\alpha'_j + \frac{1}{2}\alpha'_l\right) - \log\Gamma\left(\frac{1}{2}\beta'_j + \frac{1}{2}\beta'_l\right) \\
& - \log\left(\frac{1}{2}\alpha_j + \frac{1}{2}\alpha_l\right) + \log\Gamma\left(\frac{1}{2}\Big(s_j + n + s_l + n\Big)\right) + \log\Gamma\left(\frac{1}{2}\left(\alpha'_j + \beta'_j + \alpha_l + \beta_l\right)\right)\Bigg] \\
= & \left\{\Gamma\left(\frac{1}{2}\Big(\sum_{d=1}^{D}\alpha_{jd} + n + \sum_{d=1}^{D}\alpha_{ld} + n\Big)\right) + \Gamma\left(\frac{1}{2}\left(\alpha'_j + \beta'_j + \alpha_l + \beta_l\right)\right)\right. \\
& \left.\sqrt{\Gamma\Big(\sum_{d=1}^{D}\alpha_{jd}\Big)\Gamma\Big(\sum_{d=1}^{D}\alpha_{jl}\Big)\Gamma\Big(\alpha'_j\Big)\Gamma\Big(\alpha'_l\Big)\Gamma\Big(\beta'_j\Big)\Gamma\Big(\beta'_l\Big)\alpha_j\alpha_l}\right\} \Bigg/ \\
& \left\{\Gamma\left(\sum_{d=1}^{D}\left(\frac{1}{2}\left(\alpha_{jd} + \alpha_{ld}\right)\right)\right)\Gamma\left(\frac{1}{2}\left(\alpha'_j + \alpha'_l\right)\right)\Gamma\left(\frac{1}{2}\left(\beta'_j + \beta'_l\right)\right)\left(\frac{1}{2}\left(\alpha_j + \alpha_l\right)\right)\right. \\
& \left.\sqrt{\Gamma(\sum_{d=1}^{D}\alpha_{jd} + n)\Gamma(\sum_{d=1}^{D}\alpha_{ld} + n)\Gamma(\alpha'_j + \beta'_j)\Gamma(\alpha'_l + \beta'_l)}\right\}. \qquad (5.33)
\end{aligned}$$

Appendix 3: The KL-divergence for EMBL

The KL-divergence between two exponential distributions is given by [174]:

$$KL(p(X|\theta_j), p(X|\theta_l)) = \Phi(\theta_j) - \Phi(\theta_l) + [G(\theta_j) - G(\theta_l)]^{tr} E_{\theta_j}[T(X)] \qquad (5.34)$$

where E_θ is the expectation with respect to $p(X|\theta_j)$. Moreover, we have the following [67]:

$$E_\theta[T(X)] = -\Phi'(\theta_j) \qquad (5.35)$$

Thus, according to Eq.(5.6), we have:

$$E_{\theta_j}\Big[\sum_{d=1}^{D} I(x_d \geq 1)\Big] = -\frac{\partial \Phi(\theta_j)}{\partial \alpha_d} = \Psi(\sum_{d=1}^{D} \alpha_{jd} + n) - \Psi(\sum_{d=1}^{D} \alpha_{jd}) \tag{5.36}$$

where $n = \sum_{d=1}^{D} x_d$, and $\Psi(.)$ is the digamma function. By substituting the previous two equations into Eq.(5.34), we obtain:

$$\begin{aligned}
KL(p(X|\theta_j), p(X|\theta_l)) &= \log\Gamma\Big(\sum_{d=1}^{D}\alpha_{jd}\Big) - \log\Gamma\Big(\sum_{d=1}^{D}\alpha_{ld}\Big) + \log\Gamma(\alpha'_j) - \log\Gamma(\alpha'_l) \\
&+ \log\Gamma(\beta'_j) - \log\Gamma(\beta'_l) + \log(\alpha_j) - \log(\alpha_l) - \log\Gamma\Big(\sum_{d=1}^{D}\alpha_{jd} + n\Big) + \log\Gamma\Big(\sum_{d=1}^{D}\alpha_{ld} + n\Big) \\
&- \log\Gamma(\alpha'_j + \beta'_j) + \log\Gamma(\alpha'_l + \beta'_l) + \sum_{d=1}^{D}\left(\Psi(\sum_{d=1}^{D}\alpha_{jd} + n) - \Psi(\sum_{d=1}^{D}\alpha_{jd})\right)(\alpha_{jd} - \alpha_{ld}) \\
&= \log\left[\frac{\Gamma\Big(\sum_{d=1}^{D}\alpha_{jd}\Big)\Gamma\Big(\sum_{d=1}^{D}\alpha_{ld} + n\Big)\Gamma(\alpha'_l + \beta'_l)\Gamma(\alpha'_j)\Gamma(\beta'_j)(\alpha_j)}{\Gamma\Big(\sum_{d=1}^{D}\alpha_{ld}\Big)\Gamma\Big(\sum_{d=1}^{D}\alpha_{jd} + n\Big)\Gamma(\alpha'_j + \beta'_j)\Gamma(\alpha'_l)\Gamma(\beta'_l)(\alpha_l)}\right] \\
&+ \sum_{d=1}^{D}\left(\Psi(\sum_{d=1}^{D}\alpha_{jd} + n) - \Psi(\sum_{d=1}^{D}\alpha_{jd})\right)(\alpha_{jd} - \alpha_{ld}).
\end{aligned} \tag{5.37}$$

Appendix 4: Reńyi Kernel for EMBL

In the case of the EMBL distribution, we can show that:

$$\begin{aligned}
\int_0^{+\infty} p(\mathbf{X}|\theta_j)^{\sigma} p(\mathbf{X}|\theta_l)^{1-\sigma} dX &= \left[\frac{\Gamma(\sum_{d=1}^{D}\alpha_{jd})\Gamma(\alpha'_j)\Gamma(\beta'_j)\alpha_j}{\Gamma(\sum_{d=1}^{D}\alpha_{jd} + n)\Gamma(\alpha'_j + \beta'_j)}\right]^{\sigma} \\
&= \left[\frac{\Gamma(\sum_{d=1}^{D}\alpha_{ld})\Gamma(\alpha'_l)\Gamma(\beta'_l)\alpha_l}{\Gamma(\sum_{d=1}^{D}\alpha_{ld} + n)\Gamma(\alpha'_l + \beta'_l)}\right]^{1-\sigma} \\
&\times \int_0^{+\infty} \frac{n!}{\prod_{d=1}^{D} x_d}\left(\prod_{d=1}^{D}(\alpha_{jd})^{\sigma}(\alpha_{ld})^{1-\sigma}\right)
\end{aligned} \tag{5.38}$$

We have the PDF of an EMBL distribution that integrates to one which gives:

$$\int_0^{+\infty} \frac{n!}{\prod_{d=1}^{D} x_d}\prod_{d=1}^{D}(\alpha_{jd}) = \frac{\Gamma(\sum_{d=1}^{D}\alpha_{jd} + n)\Gamma(\alpha'_j + \beta'_j)}{\Gamma(\sum_{d=1}^{D}\alpha_{jd})\Gamma(\alpha'_j)\Gamma(\beta'_j)\alpha_j} \tag{5.39}$$

By substituting Eq.(5.39) into Eq.(5.38), we obtain:

$$\int_0^{+\infty} p(\mathbf{X}|\theta_j)^{\sigma} p(\mathbf{X}|\theta_l)^{1-\sigma} dX = \\ \left[\frac{\Gamma(\sum_{d=1}^{D}\alpha_{jd})\Gamma(\alpha'_j)\Gamma(\beta'_j)\alpha_j}{\Gamma(\sum_{d=1}^{D}\alpha_{jd}+n)\Gamma(\alpha'_j+\beta'_j)}\right]^{\sigma}\left[\frac{\Gamma(\sum_{d=1}^{D}\alpha_{ld})\Gamma(\alpha'_l)\Gamma(\beta'_l)\alpha_l}{\Gamma(\sum_{d=1}^{D}\alpha_{ld}+n)\Gamma(\alpha'_l+\beta'_l)}\right]^{1-\sigma} \\ \times \frac{\Gamma(\sum_{d=1}^{D}\sigma\alpha_{jd}+n)\Gamma(\alpha'_j+\beta'_j)}{\Gamma(\sum_{d=1}^{D}\sigma\alpha_{jd})\Gamma(\alpha'_j)\Gamma(\beta'_j)\alpha_j} \times \frac{\Gamma(\sum_{d=1}^{D}(1-\sigma)\alpha_{ld}+n)\Gamma(\alpha'_l+\beta'_l)}{\Gamma(\sum_{d=1}^{D}(1-\sigma)\alpha_{ld})\Gamma(\alpha'_j)\Gamma(\beta'_l)\alpha_l} \tag{5.40}$$

Appendix 5: Shannon Entropy for EMBL

$$H[p(\mathbf{X}|\theta_j)] = -\int_0^{+\infty} p(\mathbf{X}|\theta_j)\Bigg[\log\Gamma(\sum_{d=1}^{D}\alpha_{jd}) + \log\Gamma(\alpha'_j) + \log\Gamma(\beta'_j) + \log(\alpha_j) \\ - \log\Gamma(\sum_{d=1}^{D}\alpha_{jd}+n) - \log\Gamma(\alpha'_j+\beta'_j) + \sum_{d=1}^{D}\Big(\log(\alpha_{jd})\Big)E_{\theta}[I(x_{id}\geq 1)]\Bigg] \tag{5.41}$$

By substituting Eq.(5.36) into the previous equation, we obtain the following:

$$H[p(\mathbf{X}|\theta_j)] = -\log\Gamma(\sum_{d=1}^{D}\alpha_{jd}) - \log\Gamma(\alpha'_j) - \log\Gamma(\beta'_j) - \log(\alpha_j) + \log\Gamma(\sum_{d=1}^{D}\alpha_{jd}+n) \\ + \log\Gamma(\alpha'_j+\beta'_j) - \sum_{d=1}^{D}\log(\alpha_{jd})\Big(\psi(\sum_{d=1}^{D}\alpha_{jd}+n) - \psi(\sum_{d=1}^{D}\alpha_{jd})\Big) \tag{5.42}$$

Chapter 6

Sparse Count Data Clustering Using an Exponential Approximation to Generalized Dirichlet Multinomial Distributions

Clustering frequency vectors is a challenging task on large datasets considering its high dimensionality and sparsity nature. Generalized Dirichlet Multinomial (GDM) distribution is a competitive generative model for count data in terms of accuracy, but its parameters estimation process is slow. The exponential-family approximation of the multivariate Polya distribution has shown to be efficient to train and cluster data directly, without dimensionality reduction. In this paper, we derive a new family of distributions that approximates the GDM distributions, and we call it (EGDM). A mixture model is developed based on the new exponential family of distributions, and its parameters are learned through the Deterministic Annealing Expectation-Maximization (DAEM) approach as a new clustering algorithm for count data. Moreover, we propose the use of the Minimum Message Length (MML) criterion for selecting the optimal number of components to best describe the data with a finite EGDM mixture model. A set of empirical experiments, which concern text, image, and video clustering, has been conducted to evaluate the proposed approach performance. Results show that the new model attains a superior performance, and it is considerably faster than the corresponding method for GDM distributions.

6.1 Introduction

Count data appear in many domains in machine learning and computer vision applications. Consider, for example, text documents clustering, or image database summarization where each document or image is represented by a vector corresponding to the appearance frequencies of words or visual words, respectively. Real texts systematically exhibit the burstiness phenomenon, *i.e.*, if a word appears once in a document, it is much more likely to appear again [7, 54]. Indeed, this phenomenon is not limited to text and can also be observed in images with visual words [96]. Moreover, in bag-of-words, or bag-of-visual-words, representation, many features occur only once, and many more do not occur at all, as each observation contains only a small subset of the vocabulary. This is referred to as the sparsity nature resulting in many of the entries being zero. Thus, text documents and images are represented as high-dimensional and sparse vectors, a few thousand dimensions with a sparsity of 95 to 99% [81]. The sparseness of data is heavily studied in the literature, where many techniques have been proposed to optimize data representation for a more efficient and accurate clustering [229].

Hierarchical Bayesian modeling frameworks have the ability to model the dependency of word repetitive occurrences "burstiness". In such frameworks, the Dirichlet distribution is usually used as a conjugate prior distribution for the multinomial, which has numerous computational advantages [55]. The resulting model is the Dirichlet Compound Multinomial (DCM) [9]. The hierarchical approach of DCM considers the count vector for each document, or image, to be generated by a multinomial distribution in which parameters are generated by the Dirichlet distribution. The hierarchical Bayesian model called Generalized Dirichlet Multinomial (GDM), that is the composition of the generalized Dirichlet distribution and the multinomial, is an interesting alternative to the DCM. Indeed, several limitations of the Dirichlet can be handled by using the generalized Dirichlet distribution, which has many convenient properties that make it more useful and practical, as a prior to the multinomial, than the Dirichlet in real-life applications [10, 100]. However, the estimation procedure for GDM is very inefficient when the collection size is large. Thus, the present paper proposes that the GDM distribution can be approximated as a member of the exponential family of distributions to reduce the computation in very high-dimensional spaces.

This research work is motivated by the fact that GDM shares similar problems to the ones with DCM, including that it does not belong to the exponential family, its expression lacks intuitiveness, and its parameters cannot be estimated quickly. The author in [12] has shown that the estimation algorithm of the exponential-family approximation to the DCM, EDCM, is much faster than the corresponding algorithm based on DCM. Moreover, it models the burstiness well even for rare words, which has been indicated by the lower perplexity always achieved when using the EDCM mixtures. Thus, we derive a new distribution that is a close approximation to the GDM. The proposed distribution is a member of the exponential family of distributions that we called EGDM. Furthermore, we developed a clustering framework via a mixture of EGDMs. For learning the parameters of an EGDM mixture, we propose the use of the Deterministic Annealing Expectation-Maximization (DAEM) algorithm to avoid the initialization dependency problem of the standard EM. By means of real-life applications, we show that the DAEM algorithm with EGDM distribution is a competitive algorithm for clustering high-dimensional and sparse count data efficiently. Moreover, as the model selection is a crucial issue in mixture modeling [40, 41], we develop an MML criterion to determine the number of components that best describes the data in a finite EGDM mixture. Based on the EGDM mixture and the MML criterion, we proposed a probabilistic model for different challenging clustering tasks, namely, text documents modeling, image database categorization, and human action recognition.

The structure of the rest of this paper is organized as follows. First, Section 6.2 presents some related works and the motivation for this research. Section 6.3 reviews the Generalized Dirichlet Multinomial (GDM) distribution and its properties. Next, Section 6.4 discusses the GDM approximation in detail where we derive a new family of distributions that we called EGDM. Section 6.5 applies DAEM to learn EGDM mixture parameters and proposes an MML criterion for selecting the optimal number of components. Section 6.6 demonstrates the capabilities of the proposed approach in several applications and provides powerful evidence of the EGDM performance. Finally, Section 6.7 gives the concluding remarks.

6.2 Related Works and Motivation

Exponential families of distributions offer several appealing statistical and computational properties [65]. For instance, sufficiency retains the essential information in a dataset regarding the parameters which reduce the computation time, especially for sparse high-dimensional data. Elkan [12] proposed EDCM as an efficient approximation to the Dirichlet compound multinomial (the multivariate Polya distribution) [9]. EDCM models address the burstiness phenomenon successfully, and they are computationally faster than DCM, especially when dealing with sparse and high-dimensional vectors. EDCM has been used later to improve the modeling accuracy of different fields (for example, [230, 231]).

Despite the fact that Dirichlet distribution is flexible and has several interesting properties such as the estimation consistency, it is a conjugate prior to the multinomial, and its simplicity, it has several limitations. For instance, in the case of positively correlated data, the use of Dirichlet distribution is inappropriate, given that it has a very restrictive negative covariance structure. Another limitation of the Dirichlet distribution includes that the variables with the same mean must have the same variance [100]. All these disadvantages can be handled by using the generalized Dirichlet distribution, which is a more suitable prior to the multinomial, than the Dirichlet in real-life applications given that it has many convenient properties. The composition of the generalized Dirichlet distribution and the multinomial, introduced by Bouguila [10], is a more flexible and efficient alternative to the DCM. Indeed, the Generalized Dirichlet Multinomial (GDM) has shown to be an effective model that captures the burstiness and achieves high clustering accuracy in different applications such as image database summarization, handwritten digit recognition, text document clustering, and consumption behavior prediction [10, 175, 232]. GDM has shown a success also in regression models given its ability to learn the complex correlation between counts [233]. Hence, it would be interesting to approximate GDM distribution as a member of the exponential family of distributions to reduce the computation in very high-dimensional spaces.

On the other hand, model-based clustering involves assigning the cluster membership probabilistically as the model tries to fit the data that are assumed to be coming from a mixture of probability distributions [234, 235]. Thus, a particular clustering method is supposed to work well when

the model best fits the dataset. One of the challenges in cluster analysis is determining the number of clusters that best describes the data. This issue has been discussed in [3, 176, 236]. Several information-theory based approaches have been proposed in the literature, including the Minimum Message Length (MML) [1, 56], Akaike's Information Criterion (AIC) [57], the Minimum Description Length (MDL) [58], the Mixture of MDL (MMDL) [1]. A detailed survey of selection criteria methods can be found in [3].

Among the different proposed approaches, MML and MDL have been found to give the same result for Gaussian distributions [182], (a comparison between them was conducted in [71, 237]). However, the MML criterion has shown better results compared to the AIC and MDL criteria for artificial mixtures of Gaussians [40]. MML has been used with good results in the case of many mixture models [60]. Specifically, it has been implemented in the case of Gaussian, Poisson, and von Mises circular mixtures [41], also in spatially correlated classes of Gaussian distributions [238], and recently for discrete data with a mixture of Multinomials [239]. Moreover, recent work has shown that an MML-based approach with a finite EDCM mixture offers strong modeling capabilities for many real-world applications that involve high-dimensional count data [22].

6.3 The Generalized Dirichlet Multinomial Distribution

The Generalized Dirichlet (GD) distribution was introduced in [240], and it is mainly motivated by limitations of the Dirichlet distribution in modeling the covariances. Indeed, the variables in a Dirichlet random vector are all negatively correlated, and this is called a negative-correlation requirement. Moreover, in Dirichlet distribution, there is only one degree of freedom (by selecting the value of shape parameter), which is used to adjust the spread of a Dirichlet prior. Thus, adding individual variance information for each entry of the random vector is not possible [241]. Furthermore, additional strenuous constraints are set on the variances and the covariances in case of using the mean probabilities to solve the parameters of a Dirichlet distribution [242, 243]. Another limitation of Dirichlet distribution is the equal-confidence requirement [100]. Generally, a random variable with a small normalized variance is less uncertain than a random variable with a large normalized variance. However, the normalized variance for all variables in a Dirichlet random vector will be

the same. The Dirichlet distribution, despite these limitations, is commonly used as a prior to the Multinomial because of its computational efficiency. The generalized Dirichlet distribution, in fact, can release the constraints of the Dirichlet distribution; thus, it has shown to be a more appropriate prior for naive Bayesian classifiers [10, 100]. Moreover, the independence property of GD distribution, defined by the ability to sample each entry of the random vector from independent Beta distributions, provides more flexibility than the Dirichlet distribution [241].

In W-dimensional space, the generalized Dirichlet distribution with parameters $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_W)$, and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_W)$, is defined as [100]:

$$\mathcal{GD}(\boldsymbol{\rho}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{w=1}^{W} \frac{\Gamma(\alpha_w + \beta_w)}{\Gamma(\alpha_w)\Gamma(\beta_w)} \boldsymbol{\rho}_w^{\alpha_w - 1} \Big(1 - \sum_{l=1}^{w} \boldsymbol{\rho}_l\Big)^{\gamma_w} \tag{6.1}$$

where $0 < \rho_w < 1$, $\gamma_w = \beta_w - \alpha_{w+1} - \beta_{w+1}$, for $w = 1, \ldots, W-1$, and $\gamma_W = \beta_W - 1$. The mean and the variance of the generalized Dirichlet distribution satisfy the following conditions [240, 244]:

$$E(P_w) = \frac{\alpha_w}{\alpha_w + \beta_w} \prod_{l=1}^{w-1} \frac{\beta_l}{\alpha_l + \beta_l}, \tag{6.2}$$

$$Var(P_w) = E(P_w)\left(\frac{\alpha_w + 1}{\alpha_w + \beta_w + 1} \prod_{l=1}^{w-1} \frac{\beta_l + 1}{\alpha_l + \beta_l + 1} - E(P_w)\right), \tag{6.3}$$

and the covariance between P_{w1} and P_{w2} is:

$$\begin{aligned} Cov(P_{w1}, P_{w2}) =& E(P_{w2})\Bigg(\frac{\alpha_{w1}}{\alpha_{w1} + \beta_{w1} + 1} \\ & \prod_{l=1}^{w_1 - 1} \frac{\beta_l + 1}{\alpha_l + \beta_l + 1} - E(P_{w1})\Bigg) \end{aligned} \tag{6.4}$$

The correlations between counts can be positive or negative. Thus, the generalized Dirichlet distribution can release the negative-correlation requirement [243]. Moreover, the generalized Dirichlet includes the Dirichlet as a special case, by taking $\beta_w = \alpha_{w+1} + \beta_{w+1}$. That is, the variables in a generalized Dirichlet vector can have different normalized variances, and the GD will be reduced to a Dirichlet distribution only when all variables have the same normalized variance, which can release the equal-confidence requirement [243].

Similar to the Dirichlet, the generalized Dirichlet is also a conjugate to the multinomial distribution, but it is more flexible for several applications given that it remains W degrees of freedom [245]. The composition of the generalized Dirichlet and the multinomial gives the GDM [10]. Define $\mathbf{X} = (x_1, \ldots, x_{W+1})$ as a sparse vector of counts representing a document, or an image, where x_w corresponds to the frequency of the appearance of a word, or visual word, w, the GDM distribution is given by:

$$\mathcal{GDM}(\mathbf{X}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \frac{\Gamma(n+1)}{\prod_{w=1}^{W+1}\Gamma(x_w+1)} \prod_{w=1}^{W} \frac{\Gamma(\alpha_w+\beta_w)}{\Gamma(\alpha_w)\Gamma(\beta_w)} \prod_{w=1}^{W} \frac{\Gamma(\alpha'_w)\Gamma(\beta'_w)}{\Gamma(\alpha'_w+\beta'_w)}, \tag{6.5}$$

where $n = \sum_{w=1}^{W+1} x_w$, $\alpha'_w = \alpha_w + x_w$, and $\beta'_w = \beta_w + x_{w+1} + \cdots + x_{W+1}$, for $w = 1, \ldots, W$. It is important to note that the generalized Dirichlet is a tree of Beta distributions, and the GDM is a tree of 2-D DCMs [232]. Moreover, it is worth mentioning the generation of a document (or an image) using GDM is done in the following way: a sample is drawn from the generalized Dirichlet distribution to generate a Multinomial distribution, and then a document (or an image) is generated by the multinomial distribution. The GDM density function is thus, obtained by integrating over all possible Multinomials.

6.4 Exponential-Family Approximation to GDM

In this section, we derive a new family of distributions that is an approximation to the GDM. We called the proposed approximation EGDM, and it is, unlike the GDM, a member of the exponential family.

6.4.1 The Exponential Family of Distributions

The exponential family, a unified family of distributions, is practically convenient and widely used in spaces parameterized by finite dimensional vectors. The reasons for its popularity, specially in machine learning, include a number of important and useful calculations in statistics that contribute to both convenience and larger scale understanding [14]. A K-parameter exponential

distribution for random variables $\mathbf{X}$, can be written as:

$$P(\mathbf{X}|\theta) = h(x) \exp\Big\{ \sum_{l=1}^{K} \Phi_l(\theta) f_l(x) - g(\theta) \Big\} \tag{6.6}$$

where $\Phi_l(\theta)$ is called the natural parameter, $f_l(x)$ is the sufficient statistic, $h(x)$ is the underlying measure and $g(\theta)$ is called log normalizer which ensures that the distribution integrates to one [65].

A sufficient statistic is supposed to contain by itself all of the information about the unknown parameters that the entire sample could have provided. In other words, sufficiency characterizes what is essential in a dataset, or alternatively, what is inessential and can, therefore, be thrown away to reduce the computation time [190]. This captures the intuitive notion that $f_l(x)$ contains all of the essential information in X regarding θ_l. Thus, we retain only the sufficient statistic for the purpose of estimating the parameters [13, 14]. Any family of distributions where the support depends on the parameter can not be from an exponential family. However, it can be reduced to a member of the exponential families via a suitable transformation and re-parameterization.

6.4.2 Approximating the GDM

Given the sparsity of datasets represented using bag-of-words, or bag-of-visual-words, it should be possible to approximate the probability as a function of non-zero x_w values only for computational efficiency. That is, when $x_w = 0$ the value $\Gamma(x_w + 1) = 1$. The GDM distribution, in this case, is given by:

$$\mathcal{GDM}(\mathbf{X}) = \frac{\Gamma(n+1)}{\prod_{w:x_w \geq 1} \Gamma(x_w + 1)} \prod_{w:x_w \geq 1} \frac{\Gamma(\alpha_w + \beta_w)}{\Gamma(\alpha_w)\Gamma(\beta_w)} \prod_{w:x_w \geq 1} \frac{\Gamma(\alpha'_w)\Gamma(\beta'_w)}{\Gamma(\alpha'_w + \beta'_w)}, \tag{6.7}$$

Fitting a GDM by a maximum likelihood to a set of observations, we found experimentally that $\alpha_w \ll \beta_w \ll 1$ for almost all words w based on different datasets (see Section 6.6). In case of high dimensional data, where the parameters are really small, it is useful to use the following fact for $x \geq 1$ [12]:

$$\lim_{\alpha \to 0} \frac{\Gamma(\alpha + x)}{\Gamma(\alpha)} - \alpha\Gamma(x) = 0, \tag{6.8}$$

As α_w is much less than β_w, we can use the previous fact to replace $\Gamma(\alpha_w + \beta_w)$ in Eq.(6.7)

with $\alpha_w\Gamma(\beta_w)\Gamma(\alpha_w)$. Moreover, using the fact that if x is an integer then $\Gamma(x) = (x-1)!$, we can approximate the GDM and rewrite its density function as:

$$\mathcal{GDM}(\mathbf{X}) \approx \frac{n!}{\prod\limits_{w:x_w\geq 1} x_w!} \prod_{w:x_w\geq 1} \alpha_w \prod_{w:x_w\geq 1} \frac{\Gamma(\alpha'_w)\Gamma(\beta'_w)}{\Gamma(\alpha'_w+\beta'_w)}, \tag{6.9}$$

We reduce this approximated form of GDM to a member of exponential family using some properties of logarithm and Gamma function (see Appendix 1) to obtain the new distribution, that we call (EGDM), in the exponential family form as:

$$\begin{aligned}\mathcal{EGDM}(\mathbf{X}) = \Big(\prod_{w:x_w\geq 1} x_w\Big)^{-1} \prod_{w:x_w\geq 1} \frac{\Gamma(z_w)}{\Gamma(x_w+z_w)} n! \\ \times \exp\Big[\sum_{w=1}^{W} I(x_w\geq 1)\log\frac{\alpha_w\beta_w}{(\alpha_w+\beta_w)}\Big].\end{aligned} \tag{6.10}$$

where $I(x_w \geq 1)$ is an indicator that represents whether a word w appears at least once in the vector $\mathbf{X}$.

Besides the desirable computational properties obtained by having the distribution in exponential family form, as discussed earlier, the proposed EGDM (Eq. 6.10) has an advantage of avoiding the complications of evaluating the Gamma function and its derivatives in estimating the parameters of the original GDM (Eq. 6.5). In addition, this form shows that following EGDM multiple appearances of the same word are allowed to have a higher probability. Furthermore, this form supports modeling both frequencies of natural languages, word types, and word tokens, which is beneficial for capturing the statistical properties [66]. Similar to DCM and EDCM, the maximum likelihood estimates of GDM and EGDM are sensitive to which words appear in which documents, while Multinomial ignores the type-token distinction (*i.e.,* the Multinomial parameters are the same regardless documents boundaries in the collection).

6.5 Estimation and Selection for a Finite mixture of EGDMs

In this section, we discuss the proposed DAEM algorithm for estimating the EGDM mixture model parameters. Afterward, we develop an MML criterion for determining the optimal number

of components in the EGDM mixture and give the complete algorithm for estimation and selection.

6.5.1 Maximum Likelihood Estimation

In mixture modeling, the data are assumed to be generated from a mixture of sub populations. Let $\mathcal{X}$ to be an observed dataset with N data instances $\mathcal{X} = \{X_1, \ldots, X_N\}$, where $\mathbf{X}_i = (x_{i1}, \ldots, x_{iW+1})$ is drawn from a superposition of M EGDM densities of the form:

$$P(\mathbf{X}_i|\pi, \theta) = \sum_{j=1}^{M} \pi_j \, \mathcal{EGDM}(\mathbf{X}_i|\theta_j). \tag{6.11}$$

where π_j ($0 < \pi_j < 1$ and $\sum_{j=1}^{M} \pi_j = 1$) are the mixing proportions. Each $\mathcal{EGDM}(\mathbf{X}|\theta_j)$ represents a mixture component j that has its own parameters $\theta_j = \{\alpha_j, \beta_j\}$, where $\boldsymbol{\alpha_j} = (\alpha_{j1}, \ldots, \alpha_{jW})$, and $\boldsymbol{\beta_j} = (\beta_{j1}, \ldots, \beta_{jW})$.

For every observed data point $\mathbf{X}_i$, there is a corresponding latent variable $\mathbf{Z}_i = (z_{i1}, \ldots, z_{iM})$. The set $\mathcal{Z} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_N\}$ denotes the missing group-indicator vectors for data elements in the jth cluster. The value of z_{ij} satisfies $z_{ij} \in \{0, 1\}$, such that a particular element z_{ij} is equal to one, and all other elements are equal to 0. The complete data are considered to be $(\mathcal{X}, \mathcal{Z}|\Theta)$, where Θ is the set of all latent variables and parameters. The complete data log-likelihood corresponding to a mixture model, with M components, is given by:

$$\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta) = \sum_{i=1}^{N} \sum_{j=1}^{M} z_{ij} \, \Big(\log P(\mathbf{X}_i|\theta_j) + \log \pi_j \Big). \tag{6.12}$$

For learning a mixture model, the Expectation-Maximization (EM) algorithm is the most popular approach, which generates a sequence of models with non-decreasing log-likelihood on the data. Researchers proposed different extensions to overcome the problems associated with EM. Among the successful extensions, the deterministic annealing method (DAEM) [53] has been efficiently used to avoid initialization dependency and poor local maxima. Some interesting justifications about using the deterministic annealing procedure include that in the annealing process, the function is smoothed to have only one global optimum point by beginning at a high temperature. As the temperature decreases, the function shape gradually approaches the original objective, so the

DAEM continually tracks the new optimum point until it finds the best one. Moreover, exploring a larger region of parameter space through the slow EM convergence is an important factor in the good performance of soft clustering algorithms [110]. Practically, slower convergence makes the weights z_{ij} further away from zero and one, thus they reflect the membership uncertainty more realistically [12].

The deterministic annealing approach uses multiple phases, each with a value of computational temperature parameter. Each phase in the deterministic annealing approach runs the EM algorithm until convergence, where the final estimated model parameters Θ in each phase are used as initial values in the next one. In EM, the estimation of the parameters is done by iteratively proceeding two steps, E-step and M-step, using the notion of incomplete data, which produces a sequence of estimates $\{\Theta^{(t)}, t = 0, 1, 2, \dots\}$. When applying the deterministic annealing procedure, the posterior probabilities will be computed in the **E-step** as:

$$\hat{z}_{ij}^{(t)} = \frac{\left(P(\mathbf{X}_i|\theta_j^{(t)})\ \pi_j^{(t)}\right)^{\tau}}{\sum_{j=1}^{M}\left(P(\mathbf{X}_i|\theta_j^{(t)})\ \pi_j^{(t)}\right)^{\tau}} \tag{6.13}$$

where $\tau = \frac{1}{T}$, and T corresponds to the computational temperature. In the **M-step**, the parameters estimates will be updated according to:

$$\hat{\Theta}^{(t+1)} = \arg\max_{\Theta} \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta, \Theta^{(t)}). \tag{6.14}$$

when maximizing (6.14), we obtain:

$$\hat{\pi}_j^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N} \hat{z}_{ij}^{(t)}. \tag{6.15}$$

The maximum likelihood parameter estimate is obtained by taking the derivative of the log-likelihood function and find Θ when the derivative is equal to zero. However, a closed-form solution for the α_j and β_j parameters does not exist, thus, we use the Newton-Raphson method such that:

$$\hat{\theta}_j^{(t+1)} = \theta_j^{(t)} - H\left(\theta_j^{(t)}\right)^{-1} \frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta^{(t)})}{\partial \theta_j^{(t)}}, \tag{6.16}$$

where $H\left(\theta_j^{(t)}\right)^{-1}$ is the inverse of the Hessian matrix which is based on the second-order derivatives (see Appendix 2). Each update to the parameters resulting from the E step followed by the M step is guaranteed to increase the log-likelihood function. Hence, the algorithm is deemed to have converged when the change in the log-likelihood function, becomes insignificant.

6.5.2 MML Criterion for EGDM

Let $\mathcal{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$ be a set of data controlled by a mixture of EGDM distributions with parameters $\Theta = \{\theta_1, \ldots, \theta_M\}$, where M is the number of clusters and θ_j denotes jth component set of parameters. According to information theory, the candidate value M is considered as an optimal number of clusters if it yields the minimum amount of information, measured in $nats$ using the natural logarithm, that is needed for transmitting the dataset $\mathcal{X}$ efficiently from a sender to a receiver [71]. In case of mixture model, where the number of free parameters to be estimated is N_p, the formula for the message length is given by [40, 41]:

$$\begin{aligned} MessLength \simeq & -\log(h(\Theta)) - \log(P(\mathcal{X}|\Theta)) \\ & + \frac{1}{2}\log(|F(\Theta)|) + \frac{N_p}{2}(1 + \log(k_{N_p})) \end{aligned} \tag{6.17}$$

where $h(\Theta)$ is the prior probability, $P(\mathcal{X}|\Theta)$ is the likelihood for the complete dataset, $|F(\Theta)|$ is the determinant of the expected Fisher information matrix, and $N_p = M(2W) + 1$ is the number of free parameters for EGDM. In general, k_{N_p} is the optimal quantization lattice constant for $\mathbb{R}^{N_p}$ [72]. When $N_p = 1$ the value of $k_1 = 1/12 \simeq 0.083$, and as N_p grows, k_{N_p} tends to the asymptotic value given by $\frac{1}{2\pi e} \simeq 0.05855$ which can be approximated by $\frac{1}{12}$ [41]. The following sections show, in detail, the derivation of the MML equation proposed to determine the number of EGDM components.

6.5.2.1 Fisher Information

Fisher information of a mixture model is given by calculating the determinant of the Hessian matrix of minus complete log-likelihood [41]. In case of EGDM mixture, the complete-data Fisher information matrix has a block-diagonal structure. Generally, the prior information of different

parameters α, β and π, as well as the parameters associated with each component, are assumed to be independent. Thus, the Fisher information determinant for the complete-data is given by [1, 40]:

$$|F(\Theta)| \simeq |F(\boldsymbol{\pi})| \prod_{j=1}^{M} |F(\boldsymbol{\alpha_j}, \boldsymbol{\beta_j})| \tag{6.18}$$

We can consider the Fisher information of mixing proportions as a series of trails, where each has M possible outcomes. For each component j, the number of trails is a multinomial distribution with parameters $(\pi_1, \ldots, \pi_M)$, and the determinant $|F(\boldsymbol{\pi})|$ is, thus, given by [40]:

$$|F(\boldsymbol{\pi})| = \frac{N}{\prod_{j=1}^{M} \pi_j} \tag{6.19}$$

where N is the number of data instances.

In case of mixture models, the Fisher information matrix is usually computed after assigning the data vectors to their respective clusters [1]. Let $\mathcal{X}_j = \{\mathbf{X}_l, \ldots, \mathbf{X}_{l+\eta_j-1}\}$ be the data elements in the jth cluster where $l \leq N$ and η_j the number of the observations assigned to the jth cluster with parameters $\boldsymbol{\theta}_j$. The negative of the log-likelihood function given the set of parameters vectors $\boldsymbol{\theta_j} = \{\alpha_j, \beta_j\}$ of a single EGDM distribution can be written as:

$$\begin{aligned} -\mathcal{L}(\mathcal{X}_j|\boldsymbol{\theta}_j) &= -\log\Big(\prod_{i=l}^{l+\eta_j-1} P(\mathbf{X}_i|\theta_j)\Big) \\ &= -\sum_{i=l}^{l+\eta_j-1} \log P(\mathbf{X}_i|\theta_j) \end{aligned} \tag{6.20}$$

Each $F(\theta_j)$ is a $2W \times 2W$- block matrix, thus, we can compute the determinant of each component using the solution for block structured matrices provided in [111]. Calculating the Fisher information for a mixture of EGDMs is possible after getting the Fisher information for each single component, as following:

$$\log|F(\Theta)| = \log(N) - \sum_{j=1}^{M} \log(\pi_j) + \sum_{j=1}^{M} \log|F(\theta_j)| \tag{6.21}$$

6.5.2.2 Prior Distribution

The choice of prior distribution $h(\Theta)$ is significant as it controls the MML criterion capability. Given that other knowledge about the parameters is not available, a general independency assumption is usually made in case of mixture models where all parameters including the mixing probabilities, as prior, are independent from each other [73], such that:

$$h(\Theta) = h(\boldsymbol{\pi}) \prod_{j=1}^{M} h(\boldsymbol{\theta}_j) \tag{6.22}$$

A Dirichlet distribution with parameters $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_M)$, is a natural choice as a prior for the mixing weights vector $\boldsymbol{\pi}$ which is defined on the simplex, *i.e.,* following a Multinomial distribution. We choose a constant Dirichlet parameters (a vector of ones) which gives a uniform prior for the mixing weights, as follows [40, 41]:

$$h(\boldsymbol{\pi}) = \Gamma(M) = (M-1)! \tag{6.23}$$

For α_j, β_j, we considered similar priors to the generalized Dirichlet parameters. It has been shown experimentally in [60], that the vector θ_j can be defined on the simplex $\{(\alpha_{j1}, \beta_{j1}, \ldots, \alpha_{jW}, \beta_{jW}) : \sum_{w=1}^{W}(\alpha_{jw} + \beta_{jw}) < 2We^5\}$. Considering a symmetric Dirichlet distribution as a prior gives a uniform value as [60]:

$$h(\theta_j) = \frac{(2W)!}{(2We^5)^{2W}} \tag{6.24}$$

Substituting the prior for the mixing weights Eq.(6.23), and the prior for the parameters vector Eq.(6.24) in Eq.(6.22) gives the prior probability of a mixture of EGDM distributions. Thus, the log of the prior distribution is given by:

$$\log(h(\Theta)) = \sum_{j=1}^{M-1} \log(j) - 10MW - 2MW \log(2W) + M \sum_{w=1}^{2W} \log(w) \tag{6.25}$$

By substituting Eq.(6.25) and Eq.(6.21) into Eq.(6.17), we obtain the MML criterion for a finite mixture of EGDM distributions given a candidate value M.

In Algorithm 6, we summarize the algorithm for determining the optimal number of clusters

and learning the parameters for a finite mixture of EGDM distributions. The input to this algorithm consists of a dataset of count vectors and a set of candidate values for the number of mixture components. Its output is the number of components that best describes the data, as well as the estimated parameters corresponding to the lowest message length. For each candidate value M, we estimate the parameters using the DAEM. Experimentally, we have concluded that setting $\tau_{min} = 0.04$ is enough, and for the temperature schedule, we used $const = 5$. These reasonable choices correspond to three-phases annealing, which has shown to give good results, as explained in [10, 12].

Algorithm 6: Estimation and Selection for EGDM mixture model.

Output: The optimal components M^*, best parameters estimates Θ^*
Input: Dataset $\mathcal{X}$, with N W-dimensional vectors, and a set of candidate number of clusters $M_{min}, \ldots, M_{max}$

for $M_{min} \leq M \leq M_{max}$ **do**
  Set $\tau \leftarrow \tau_{min} (\tau_{min} \ll 1)$;
  Assign data objects to M classes using Spherical k-Means;
  Choose an initial estimate $\Theta^{(0)}$. Set $t \leftarrow 0$;
  while $\tau \leq 1$ **do**
    while *Convergance creteria is not reached* **do**
      Set $t \leftarrow t + 1$;
      for $i = 1$ ***to*** N **do**
        for $j = 1$ ***to*** M **do**
          Compute the posterior probabilities $z_{ij}^{(t)}$ using (6.13);
        end
      end
      for $j = 1$ ***to*** M **do**
        Update the mixing proportion $\pi_j^{(t)}$ using (6.15);
        Update the parameters $\theta_j^{(t)}$ using (6.16);
      end
    end
    Annealing: increase τ ($\tau \leftarrow \tau \times const.$) ;
  end
  Calculate the associated MML criterion: $MessLength(M)$ using (6.17);
end
Select the optimal Θ^*, M^* such that: $M^* = \arg\min_M MessLength(M)$;

6.6 Experimental Results

In this section, we demonstrate the effectiveness of the proposed approach via three interesting applications; text classification, image categorization, and human action recognition. In our experiments, we compare the newly proposed family of distributions EGDM with other generative models that have been used previously for count modeling and clustering. For each dataset, we run each algorithm 50 times with different random initializations. To give a baseline of the difficulty of the problem, we compare our proposed algorithm to other clustering methods such as the Spherical k-Means (SKM), the Gaussian mixture model (GMM), and the mixture of Multinomials (MM). Moreover, we compare to other generative models that have been previously proposed for modeling count data, including mixtures of Dirichlet Compound Multinomial (DCM) [9], the exponential approximation to DCM (EDCM) [12], and Multinomial Scaled Dirichlet (MSD) [23]. All the experiments are done directly (*i.e.*, we do not separate the dataset into training and testing sets). Moreover, we used the proposed MML criterion to find the optimal number of classes and evaluated the proposed MML criterion capability to select the optimal number of clusters. For validation purposes, we have considered the true number of classes as well as the true labels of each dataset as a ground truth. That is, the estimated class labels by our model were compared against the initial ones to evaluate the model's performance based on standard classification measures.

6.6.1 Text Documents Modeling

Given the rapid growth of on-line information, the demand for developing efficient methods for handling and organizing data in the textual format is becoming more crucial. Text clustering is a significant task for finding interesting information within the World Wide Web, digital libraries, and electronic mail [246]. The performance of the model is evaluated for text modeling, where we compared using precision and recall averaged at macro level [247], to assess the overall performance of the tested models across different sets of data. Moreover, we considered the mutual information (MI) [12, 110], to quantify how much the assigned classes by an algorithm agrees with the pre-specified ones. The experiments presented below are based on three datasets that have been

considered in the past (see for example, [11, 77]), namely WebKB4[1], the ModApte version of the Reuters-21578[2], and IMDB[3].

The **WebKB4** dataset is a subset of the web pages collected by the World Wide Knowledge Base (Web→Kb) project of the CMU text learning group [248]. The complete dataset gathered from computer science departments of various universities containing $8,282$ pages were manually classified into seven categories and characterized by $7,786$ features, and the average length of each document is 49.7. The considered subset is limited to the four most common categories: Course (930), Faculty (1124), Project (504), and Student (1641). **Reuters-10** is a subset of the well-known corpus Reuters-21578, contains thousands of documents collected from Reuters newswire in 1987. The complete dataset is composed of 135 classes with a vocabulary of $15,996$ words and an average document length of 192.9. The documents in this dataset are multi labeled, as they may belong to $0, 1,$, or many categories. The considered subset is composed of the 10 categories having the highest number of class members ($6,775$ and $2,258$ training and testing documents are considered for this subset, respectively). **IMDB** (movie reviews) contains positive and negative sentiments [115]. Ratings on IMDB are given as star values $\in \{1, 2, \ldots, 10\}$, which were linearly mapped to $[0, 1]$ to use as document labels; negative and positive, respectively. We used a union of the training and testing sets having around $25,000$ samples from each positive/negative group with $76,340$ unique words in total.

The pre-processing of WebKB4 involves removing all stop and rare words (less than 50 occurrences in our experiments) from the vocabularies. Rainbow package [78] is used to perform the feature selection considering words with the highest average mutual information with the class variable. Each web page is then represented using the vector space model, where each is represented as a vector containing the frequency of the occurrences of the words. Stop words have already been removed in Reuters-10 collection, so we did not remove any additional words. For sentiment analysis, certain stop words (*e.g.,* negating words) are indicative, so traditional stop word removal was not used in the IMDB dataset. Each text file is then represented as a vector containing the occurrence frequency for each word from the vocabulary.

[1] http://www.cs.cmu.edu/~webkb/
[2] http://kdd.ics.uci.edu/databases/reuters21578
[3] http://ai.stanford.edu/~amaas/data/sentiment/

Table. 6.1 indicates the average performance metrics with standard errors. The reported time (in seconds) is for a single run to the convergence of an optimized MATLAB2017a code on an Intel(R) Core(TM) i7-6700 Processor PC has the Windows 7 Enterprise Service Pack 1 operating system with a 16 GB main memory. Compared to GDM, the EGDM-based clustering is between 7 and 16 times faster for the different datasets. Likewise, the EDCM is 7 and 19 times faster than the corresponding DCM based on the considered datasets. We can notice that generative models that provide good results are generally slow, which will increase the complexity in high dimensional spaces. On the other hand, the average precision and recall achieved by EGDM clustering for WebKB4 are (88.89%) and (88.66%), for the Reuters-10 are (79.94%) and (94.74%), and finally for IMDB are (81.36%) and (89.55%). The GDM and its approximation clearly outperform all other models on all the datasets. According to a Student's t-test, the differences in performance between the proposed EGDM, and other tested models are statistically significant (p-values are between 0.0005 and 0.0037). Furthermore, the improvement achieved by EGDM over the similar approach of EDCM [12] is statistically significant, as shown by a Student's t-test (p-values are between 0.0096 and 0.0191). The mutual information gained using EGDM, and EDCM for clustering the WebKB4 dataset is (0.8957) and (0.7788), and for the Reuters-10 documents are (0.8921) and (0.7511) using EGDM and GDM, respectively. The differences here are also statistically significant. For instance, the p-values of a t-test for the difference between the mutual information gained using EGDM and other models are 0.0032 and 0.0015 for MI using EDCM, and GDM, respectively. Thus, the average accuracies, mutual information, and learning time for all the datasets confirm a competitive performance for EGDM to the widely used generative models for count data that can handle burstiness.

6.6.2 Image Database Categorization

With the exponential increase of the size of digital image collections, many computer vision tasks, including object detection, content-based image classification, and retrieval, have been well studied over the last decades. Recent successful approaches are inspired by the text retrieval systems, where images are represented as a "Bag of Visual Words" or "Bag of Features", where the local image patches are the visual equivalents of individual words [89, 249].

Table 6.1: Clustering results using EGDM mixture model for the three documents collections.

Dataset	Model	Precision	Recall	Mutual info.	time
WebKB4	SKM	30.25±0.01	29.55±0.02	0.3555±0.02	31.09
	GMM	75.83±0.03	75.30±0.02	0.7926±0.05	23.57
	MM	81.32±0.05	82.46±0.08	0.7330±0.06	17.24
	DCM	83.72±0.21	84.44±0.27	0.7651±0.43	127.96
	MSD	88.17±0.05	88.27±0.06	0.8943±0.03	44.05
	GDM	86.38±0.03	87.26±0.05	0.8420±0.18	138.26
	EDCM	84.66±0.16	84.50±0.12	0.7788±0.03	16.07
	EGDM	**88.89±0.04**	**88.66±0.05**	**0.8957±0.29**	19.26
Reuters-10	SKM	25.08±0.07	27.53±0.05	0.3392±0.02	136.03
	GMM	70.99±0.03	89.99±0.04	0.8090±0.05	220.34
	MM	72.17±0.04	91.76±0.02	0.8332±0.03	62.93
	DCM	74.84±0.05	93.56±0.02	0.8832±0.02	396.20
	MSD	75.59±0.01	90.85±0.01	0.8914±0.02	163.97
	GDM	75.66±0.02	90.86±0.03	0.7511±0.35	402.50
	EDCM	79.36±0.08	94.72±0.07	0.8820±0.45	20.68
	EGDM	**79.94±0.02**	**94.74±0.07**	**0.8921±0.29**	25.46
IMDB	SKM	55.90±0.03	55.91±0.07	0.5800±0.03	252.76
	GMM	61.40±0.05	61.40±0.04	0.6719±0.07	169.48
	MM	64.18±0.05	64.40±0.06	0.6520±0.03	143.52
	DCM	71.14±0.05	89.45±0.05	0.8578±0.09	227.11
	MSD	76.44±0.02	84.54±0.04	0.8432±0.03	254.46
	GDM	75.55±0.02	81.43±0.07	0.8992±0.05	338.26
	EDCM	78.54±0.09	89.33±0.14	0.8861±0.07	32.07
	EGDM	**81.36±0.08**	**89.55±0.04**	**0.8994±0.22**	48.26

Our baseline system builds upon the bag-of-features approach, which has demonstrated excellent performance in image classification. Bag of Features methods often rely on detecting the location and scale of localized regions from which image features are extracted. First, a set of interest points are detected and described on each patch independently using the Scale-Invariant Feature Transform (SIFT) [90]. Then, the set of descriptors is quantized using an unsupervised clustering approach into a number of homogeneous clusters as proposed in [91], where the centroid of each cluster is considered as a visual word. Each novel image is then to be represented as a histogram of frequencies corresponding to assigning the features extracted to the closest visual word using Euclidean distance.

We evaluated our model performance on three different datasets. The first one is **CIFAR-10**[4]

[4] https://www.cs.toronto.edu/~kriz/cifar.html

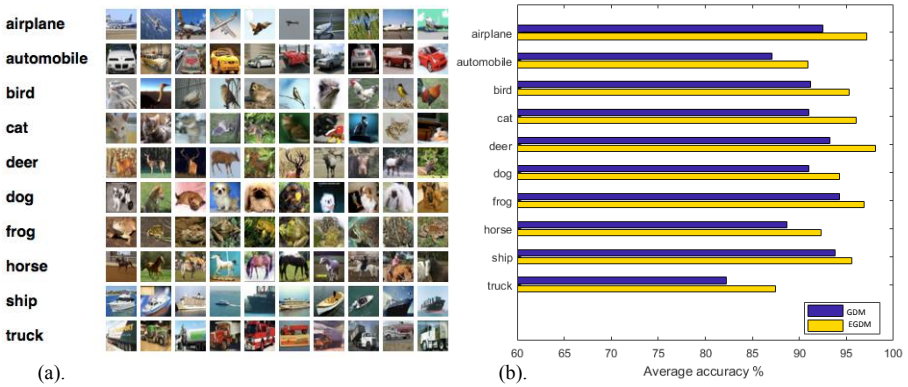


Figure 6.1: (a) Sample images from CIFAR-10 dataset. (b) Intra-class accuracy obtained by GDM vs. EGDM for CIFAR-10.

dataset which has been collected by researchers at MIT and NYU over the span of six months [92]. The dataset consists of $60,000$ natural tiny color images of size 32×32 collected from several search engines based on $79,000$ search terms. The images belonging to 10 completely mutually exclusive categories are split into $50,000$ training images and $10,000$ test images ($1,000$ images per class). Fig. 6.1 (a) presents a random sample with 10 images from each class. The second dataset is a subset of the extensive Scene UNderstanding (**SUN**) database [250], that contains 899 categories and $130,519$ images. We use $1,849$ natural scenes belonging to six categories (458 coasts, 228 river, 231 forests, 247 field, 518 mountains, and 167 sky/clouds). Fig. 6.2(a) shows example images from each class in this dataset. The last dataset is the environmental scene database by **Oliva and Torralba** [251] (refereed to as OT). This dataset is composed of about $2,688$ natural scenes classified as eight categories (360 coasts, 328 forest, 374 mountain, 410 open country, 260 highway, 308 inside of cities, 356 tall buildings, and 292 streets). The average size of each image is 250×250 pixels.

To obtain the bag-of-features representation in our experiments, we learn $25k$, $17k$, and $10k$ visual vocabulary to construct the codebook for CIFAR-10, SUN and OT dataset, respectively, and used the Euclidean distance to assign the features to the closest terms in the vocabulary resulting in frequency vectors. Table. 6.2 shows the clustering performance of the tested methods reported as the average accuracy plus/minus standard error, as well as the average run time in seconds. The accuracies achieved using EDCM and EGDM (94%) and using DCM, and GDM (90%) indicate

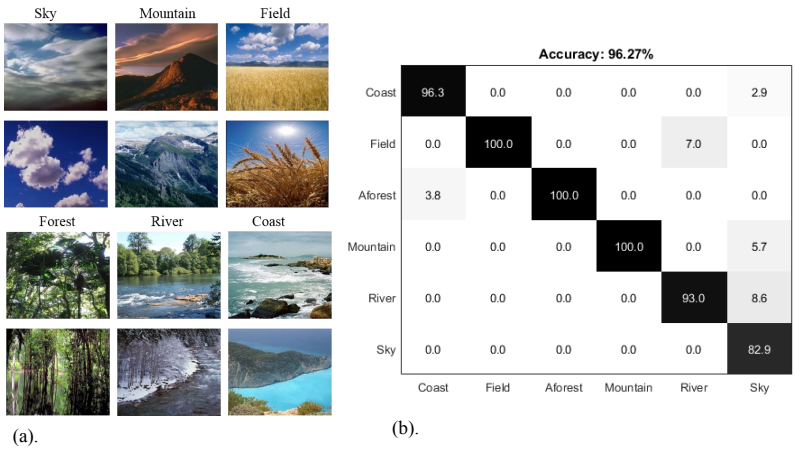


Figure 6.2: (a) Sample images from SUN dataset. (b) Confusion matrix for SUN using EGDM.

Table 6.2: The average accuracy and learning time using different methods for image categorization.

MODEL	CIFAR-10		SUN		OT	
	ACCURACY	TIME	ACCURACY	TIME	ACCURACY	TIME
SKM	24.74±0.03	174.05	25.78±0.04	45.59	29.66±0.02	59.39
GMM	36.24±0.05	450.10	37.95±0.05	69.64	35.12±0.07	47.90
MM	89.05±0.08	332.90	79.50±0.08	58.60	80.20±0.07	51.87
DCM	90.46±0.12	494.81	79.82±0.47	89.23	80.56±0.06	85.54
MSD	90.81±0.07	450.51	80.75±0.05	123.37	80.98±0.06	139.70
GDM	90.50±0.04	636.54	80.17±0.03	119.54	80.60±0.03	104.05
EDCM	94.40±0.36	108.68	93.45±0.41	44.34	80.63±0.04	28.13
EGDM	**94.41±0.02**	209.72	**96.27±0.13**	57.18	**82.77±0.02**	49.33

comparable performance. However, EGDM and EDCM outperform the corresponding GDM and DCM as the differences are statistically significant, as shown by a Student's t-test (*i.e.*, p-values between 0.0082 and 0.0014 for the different runs). Moreover, EGDM and EDCM are 3 and 5 times faster than the corresponding GDM and DCM, respectively. We can notice that DCM, MSD, and GDM behave similarly in all the tested datasets. However, the clustering accuracy achieved by GDM is very much improved by the exponential approximation approach (*e.g.,* the accuracy of categorizing the SUN dataset has increased from (80.17%) to (96.27%) using EGDM mixture). This difference is statistically significant as shown by a Students t-test (*i.e.*, p-values are between 0.0015 and 0.0035 for the different runs). The intra-class performance for the CIFAR-10 database using GDM and EGDM is shown in Fig. 6.1(b). The best classified object by EGDM is *airplane*

(with a performance of 97.20%) and *frog* by GDM (with a performance of 94.30%). The clustering accuracy has been improved for all categories, and the overall performance has a statistically significant enhancement from (90.50%) for GDM to (94.41%) for EGDM. For the SUN dataset, we can see from the confusion matrix Fig. 6.2(b), that the most difficult scenes to classify using EGDM are *Sky* and *River* with a performance of (82.9) percent and (93.0) percent, respectively. The overall accuracy of categorizing SUN using the GDM mixture has been improved from (80.17%) to (96.27%) using the EGDM clustering approach. This difference is statistically significant, according to the Student's t-test (p-values are between 0.0034 and 0.0024). As shown in Table. 6.2, the overall accuracy of categorizing OT using the GDM mixture has been improved from (80.60%) to (82.77%) using the EGDM clustering approach. Moreover, the EGDM model is noticeably faster than the corresponding GDM for categorizing both SUN and OT datasets. This result once again demonstrates the merit of using the EGDM algorithm and its superior performance for categorizing images represented as bag-of-features.

6.6.3 Human Action Recognition

With thousands of videos available due to the improvement of digital technologies, grouping them according to their contents is highly demanded to be used for organizing, summarizing, and retrieving this massive amount of data. Video classification is a challenging computer vision task with many applications. In particular, automatically recognizing human motion and activity in videos has captured scholars' interest because of its several critical applications such as motion capture, sports and entertainment analysis, monitoring and surveillance, and human-computer interaction [252]. The goal of this section is to investigate the effectiveness of the proposed approach in human action categorization in video sequences. The problem can be formulated as a clustering task for the extracted frames using the well known BoF approach, which has shown to give excellent results for human action recognition [253].

Our experiments were conducted on three publicly available datasets: Ballet [254], UCF sports [215], and YouTube [255]. For representing each dataset as BoF, the samples in each action were randomly split into 80:20 as training and testing sets. The overall accuracy for human action recognition obtained using the different generative models is shown in Table 6.3. The **Ballet** dataset

Table 6.3: The average accuracy and learning time using different methods for human action recognition.

Model	Ballet		UCF sport		Youtube	
	Accuracy	Time	Accuracy	Time	Accuracy	Time
SKM	19.60±0.04	27.56	23.40±0.02	31.47	23.79±0.04	46.23
GMM	22.03±0.02	41.79	29.20±0.02	60.23	24.11±0.06	84.73
MM	64.95±0.03	64.05	61.72±0.02	112.16	49.62±0.03	123.56
DCM	64.23±0.09	178.24	66.97±0.34	183.50	58.18±0.22	160.83
MSD	66.93±0.05	174.69	65.97±0.03	299.85	58.37±0.04	414.20
GDM	66.16±0.03	241.80	69.77±0.04	393.20	58.51±0.08	438.75
EDCM	76.43±0.22	14.37	82.69±0.02	20.12	80.40±0.06	24.67
EGDM	**79.37±0.05**	23.05	**86.73±0.06**	35.25	**83.24±0.03**	43.88

Figure 6.3: Sample frames from Ballet video dataset.

contains 44 real video sequences that consist of eight actions, such as jumping, turning, leg swinging, and standing (see examples in Fig. 6.3). From the Table, we can see that our approach achieves the highest accuracy among the compared methods. Clearly, EGDM outperforms both GDM and EDCM, which itself is better than the DCM and other models for the three tested datasets. The accuracies for clustering the Ballet dataset are (79.37%) using EGDM, (66.16%) and (76.43%) for GDM and EDCM, respectively.

The **UCF** sports dataset contains 150 real videos with the resolution of 720×480. The collection represents a natural pool of actions featured in a wide range of scenes and viewpoints with non-uniform backgrounds and moving camera/subjects. It involves 10 categories of human actions collected from different sports include Diving (14 videos), Golf Swing (18 videos), Kicking (20 videos), Lifting (6 videos), Riding Horse (12 videos), Running (13 videos), Skate Boarding (12

videos), Swing-Bench (20 videos), Swing-Side (13 videos), and Walking (22 videos). The overall accuracy for recognizing human action UCF sports dataset using EGDM mixture is (86.73%) which is significantly higher than (82.69%), by EDCM and (69.77%) by GDM mixture. Moreover, the time complexity of the EM algorithm for training EGDM and EDCM mixtures is quite fast. Comparing to GDM, the running time for EGDM is 10 to 11 times faster, and the EDCM is 6 to 9 times faster than the corresponding DCM. This will cut clustering time dramatically based on the number of clusters, the number of frames, and the dimension of the data.

The last dataset is **YouTube** contains 11 action categories: basketball shooting, biking/cycling, diving, golf swinging, horseback riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog (see samples in Fig. 6.4). This dataset is very challenging for several reasons, including large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, illumination conditions, etc. Fig. 6.5 (a) shows the comparison of the clustering accuracies using EDCM and EGDM. The overall improvement using EGDM is about 3.4% over EDCM. Most categories obtained improvement in terms of clustering accuracy except for soccer juggling (shown as *s_juggling*) and tennis swing (*t_swing*). Fig. 6.5 (b) shows the confusion table for classification using the EGDM. We can see that a lot of *t_swing* is misclassified into *s_juggling*, and *golf_swing*. Once again, the differences in performance, considering the overall accuracy, between EGDM and other models are statistically significant for all the tested datasets, as shown by a Student's t-test. For instance, EGDM performs significantly better than the earlier proposed EDCM, which itself significantly outperforms all other tested models (*i.e.,* p-values are between 0.0048 and 0.0076, for the different runs).

Indeed, the challenge to handle count data increases as the number of dimensions, classes, and data instances increases. The exponential approximation to GDM proposed in this work has shown to cluster such data faster and more efficiently compared to other generative models widely used for count data, including a similar approach that approximates DCM to the exponential family of distributions. Given the superior performance of the proposed approach in the presented applications and data examples, we can conclude that a mixture of EGDMs provides a promising clustering method for high-dimensional sparse count data.

Figure 6.4: Sample frames from YouTube Action dataset.

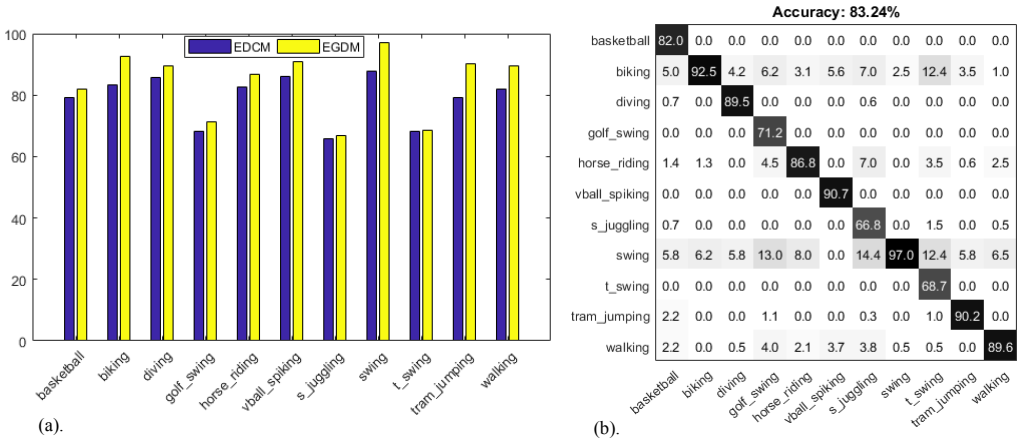


Figure 6.5: (a) Comparison of clustering performance for YouTube dataset using EDCM and EGDM, the average accuracy are 80.40% and 83.24%, respectively. (b) The confusion matrix for clustering using the proposed EGDM.

6.6.4 Model Selection Evaluation

We evaluate the performance of the proposed MML approach on all the datasets used for the different applications according to whether the selected K agrees with the prespecified number of clusters. For each dataset, we perform the model selection by running MML for a set of candidates values and consider the best candidate value K^* that minimizes the message length.

The first row of Fig. 6.6 shows the number of clusters found by the proposed criterion for WebKB4, Reuters-10, and IMDB datasets, respectively. We can see that the values of the MML criterion agree with the correct number of clusters for all text datasets. The optimal number of components found using MML with WebKB4 documents collection is K^*= 4 also agrees with the

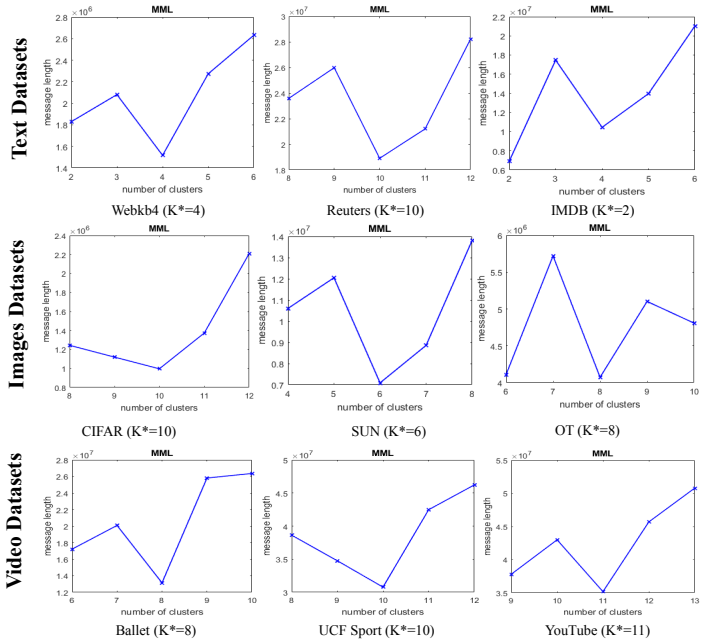


Figure 6.6: Number of clusters found by the MML criterion for the different datasets.

prespecified number of clusters. Moreover, the selected models by MML for the Reuters and IMDB datasets are K^*= 10 and K^*= 2, respectively, which corresponds to the actual numbers of clusters. The second row shows that the MML criterion is capable of selecting the optimal number of clusters to represent image datasets. The numbers of classes that yield to the minimum message length were K^*= 10, K^*=6 and K^*=8 for CIFAR-10, SUN and OT datasets, respectively, which agree with the exact prespecified number of classes. The results for the video datasets (shown at the last row of Fig. 6.6) also demonstrate the effectiveness of the proposed approach, where the correct number of clusters was found for each dataset. Thus, we can conclude that our algorithm performs well in determining the number of components that best describes the data.

6.7 Conclusion

In this work, we have introduced a new distribution that we call (EGDM) based on the exponential family approximation of the Generalized Dirichlet Multinomial (GDM) to cluster high-dimensional and sparse count data. The new model successfully and correctly captures the burstiness phenomenon, and it is many times faster and computationally efficient compared to the corresponding GDM. For learning the parameters of EGDM mixture and determining the number of optimal clusters, the Deterministic Annealing Expectation-Maximization (DAEM) algorithm and Minimum Message Length (MML) criterion have been used, respectively. The effectiveness of the proposed model was shown experimentally through demanding clustering problems involving text documents modeling, image categorization, and human action recognition in videos. The model presented in this chapter is also applicable to many other problems that involve sparse vectors of count data.

Appendix 1: Proof of Eq.(6.10)- The Exponential GDM

To reduce the approximated density of GDM (Eq.6.9) to a member of exponential family, we used some properties of logarithm including $\log(xy) = \log x + \log y$ and $\log(x/y) = \log x - \log y$, thus we obtain:

$$
\begin{aligned}
q(\mathbf{X}) = \frac{n!}{\prod\limits_{w:x_w \geq 1} x_w!} \prod_{w:x_w \geq 1} \alpha_w \exp\Big[\sum_{w:x_w \geq 1} & \log\Gamma(\alpha_w + x_w) \\
& + \log\Gamma(\beta_w + x_{w+1} + \cdots + x_{W+1}) \\
& - \log\Gamma(\alpha_w + \beta_w + x_w + x_{w+1} + \cdots + x_{W+1}) \Big],
\end{aligned} \tag{6.26}
$$

Let $z_w = x_{w+1} + \cdots + x_{W+1}$ be the cumulative sum, and using the previously mentioned fact in Eq.(6.8) we can rewrite Eq.(6.26) as:

$$q(\mathbf{X}) \approx \frac{n!}{\prod\limits_{w:x_w \geq 1} x_w!} \prod_{w:x_w \geq 1} \alpha_w \exp\Big[\sum_{w:x_w \geq 1} \Big(\log \Gamma(x_w) + \log \alpha_w \Gamma(\alpha_w) \Big) + \Big(\log \Gamma(z_w) + \log \beta_w \Gamma(\beta_w) \Big) - \Big(\log \Gamma(x_w + z_w) + \log(\alpha_w + \beta_w) \Gamma(\alpha_w + \beta_w) \Big) \Big], \tag{6.27}$$

In order to have $q(\mathbf{X})$ in the exponential family form, we need to have the support function independent of the parameters. Utilizing the logarithm and Gamma function properties mentioned above gives:

$$q(\mathbf{X}) \approx \frac{n!}{\prod\limits_{w:x_w \geq 1} x_w!} \prod_{w:x_w \geq 1} \alpha_w \exp\Big[\sum_{w:x_w \geq 1} \log \frac{(x_w - 1)!(z_w - 1)!}{(x_w + z_w - 1)!} + \frac{\alpha_w \Gamma(\alpha_w) \beta_w \Gamma(\beta_w)}{(\alpha_w + \beta_w) \Gamma(\alpha_w + \beta_w)} \Big], \tag{6.28}$$

Using Eq.(6.8) again to approximate $\Gamma(\alpha_w + \beta_w)$ gives:

$$q(\mathbf{X}) \approx \frac{\exp[\log \frac{(x_w - 1)!(z_w - 1)!}{(x_w + z_w - 1)!}]}{\prod\limits_{w:x_w \geq 1} x_w!} n! \exp\Big[\sum_{w:x_w \geq 1} \Big(\log(\alpha_w) + \log \frac{\beta_w}{(\alpha_w + \beta_w)} \Big) \Big]. \tag{6.29}$$

Rewriting Eq.(6.29) using the fact that $x! = x(x-1)!$, we can further simplify to obtain the new distribution that we call (EGDM) in the exponential family form as shown in Eq.(6.10).

Appendix 2: Newton-Raphson Approach for Estimating EGDM Parameters

The log-likelihood of one observation $\mathbf{X}_i$, following EGDM, is:

$$\log P(\mathbf{X}_i|\theta_j) = \log(n_i!) + \sum_{w:x_{iw}\geq 1} \Big(\log\Gamma(z_{iw}) - \log\Gamma(x_{iw}+z_{iw}) \Big) + \sum_{w:x_{iw}\geq 1} \Big[(\log(\alpha_{jw}) + \log(\beta_{jw}) - \log(\alpha_{jw}+\beta_{jw})) - \log(x_{iw}) \Big] \tag{6.30}$$

By computing the first derivative of $\mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta, \Theta^{(t)})$ with respect to α_{jw} and β_{jw}, we obtain:

$$\frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta, \Theta^{(t)})}{\partial \alpha_{jw}} = \sum_{i=1}^{N} z_{ij}\ I(x_{iw}\geq 1) \Big[\frac{1}{\alpha_{jw}} - \frac{1}{\alpha_{jw}+\beta_{jw}} \Big], \tag{6.31}$$

$$\frac{\partial \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta, \Theta^{(t)})}{\partial \beta_{jw}} = \sum_{i=1}^{N} z_{ij}\ I(x_{iw}\geq 1) \Big[\frac{1}{\beta_{jw}} - \frac{1}{\alpha_{jw}+\beta_{jw}} \Big]. \tag{6.32}$$

The Hessian matrix is based on the second-order derivatives, as flows:

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta, \Theta^{(t)})}{\partial \alpha_{jw1} \partial \alpha_{jw2}} = \begin{cases} \sum_{i=1}^{N} z_{ij}\ I(x_{iw}\geq 1) \Big[\frac{1}{(\alpha_{jw}+\beta_{jw})^2} - \frac{1}{\alpha_{jw}^2} \Big] & \text{if } w_1 = w_2 = w, \\ 0 & \text{otherwise,} \end{cases} \tag{6.33}$$

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta, \Theta^{(t)})}{\partial \beta_{jw1} \partial \beta_{jw2}} = \begin{cases} \sum_{i=1}^{N} z_{ij}\ I(x_{iw}\geq 1) \Big[\frac{1}{(\alpha_{jw}+\beta_{jw})^2} - \frac{1}{\beta_{jw}^2} \Big] & \text{if } w_1 = w_2 = w, \\ 0 & \text{otherwise,} \end{cases} \tag{6.34}$$

$$\frac{\partial^2 \mathcal{L}(\mathcal{X}, \mathcal{Z}|\Theta, \Theta^{(t)})}{\partial \alpha_{jw1} \partial \beta_{jw2}} = \begin{cases} \sum_{i=1}^{N} z_{ij}\ I(x_{iw}\geq 1) \Big[\frac{1}{(\alpha_{jw}+\beta_{jw})^2} \Big] & \text{if } w_1 = w_2 = w, \\ 0 & \text{otherwise,} \end{cases} \tag{6.35}$$

Then, the Hessian matrix has a block-diagonal structure:

$$H(\theta_j) = block - diag\{H_1(\alpha_{j1}, \beta_{j1}), \ldots, H_W(\alpha_{jW}, \beta_{jW})\}, \tag{6.36}$$

We remark that $H_w(\alpha_{jw}, \beta_{jw})$ can be written as:

$$H_w(\alpha_{jw}, \beta_{jw}) = D + \gamma\, \mathbf{a}\, \mathbf{a}^{tr}, \tag{6.37}$$

where:

$$D = diag\Big[\sum_{i=1}^{N} z_{ij}\; I(x_{iw} \geq 1)\frac{-1}{\alpha_{jw}^2}, \sum_{i=1}^{N} z_{ij}\; I(x_{iw} \geq 1)\frac{-1}{\beta_{jw}^2}\Big], \tag{6.38}$$

and $\gamma = \sum_{i=1}^{N} z_{ij}\; I(x_{iw} \geq 1)\Big[\frac{1}{(\alpha_{jw}+\beta_{jw})^2}\Big]$, and $\mathbf{a}^{tr} = 1$. Then, the inverse of the matrix $H_w(\alpha_{jw}, \beta_{jw})$ is given by [95], [Theorem 8.3.3]:

$$H_w(\alpha_{jw}, \beta_{jw})^{-1} = D^* + \delta^*\, \mathbf{a}^*\, \mathbf{a}^{*tr}, \tag{6.39}$$

where:

$$D^* = D^{-1} = diag[1/D_1, 1/D_2] = diag\left[\frac{-\alpha_{jw}^2}{\sum\limits_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}, \frac{-\beta_{jw}^2}{\sum\limits_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}\right], \tag{6.40}$$

$$a^{*tr} = (1/D_1, 1/D_2) = \left(\frac{-\alpha_{jw}^2}{\sum\limits_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}, \frac{-\beta_{jw}^2}{\sum\limits_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}\right), \tag{6.41}$$

$$
\begin{aligned}
\delta^* &= -\gamma(1+\gamma(1/D_1+1/D_2))^{-1} \\
&= \frac{-\sum_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}{(\alpha_{jw}+\beta_{jw})^2} \times \Bigg(1 + \Big(\frac{\sum_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}{(\alpha_{jw}+\beta_{jw})^2} \times \frac{-\alpha_{jw}^2}{\sum_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}\Big) \\
&+ \Big(\frac{\sum_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}{(\alpha_{jw}+\beta_{jw})^2} \times \frac{-\beta_{jw}^2}{\sum_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}\Big)\Bigg)^{-1} \\
&= \frac{-\sum_{i=1}^{N} z_{ij} I(x_{iw} \geq 1)}{(\alpha_{jw}+\beta_{jw})^2}\Big(1 - \frac{\alpha_{jw}^2+\beta_{jw}^2}{(\alpha_{jw}+\beta_{jw})^2}\Big)^{-1}. \qquad (6.42)
\end{aligned}
$$

Chapter 7

A Novel MM Framework for Simultaneous Feature Selection and Clustering of High-Dimensional Count Data

Count data are widely used in machine learning and computer vision applications, and they usually suffer from the well-known curse of dimensionality, which decline the performance of clustering algorithms dramatically. Feature selection is a significant technique for handling a large number of features, which most are often redundant and noisy. In this work, we propose a probabilistic approach for count data based on the concept of feature saliency in the context of mixture-based clustering using the generalized Dirichlet multinomial (GDM) distribution. By minimizing the message length, the saliency of irrelevant features is driven toward zero, which corresponds to performing feature and model selection simultaneously. Through a set of challenging applications involving text and images clustering, it is demonstrated that the developed approach performs effectively in selecting both the optimal number of clusters and the most relevant features and, thus, improve the clustering performance considerably.

7.1 Introduction

Feature selection is a traditional and practical approach to handle high-dimensional data [256]. Theoretically, a clustering algorithm supposes to perform better with the more information we have about each pattern, which seems to suggest using as many features as possible. However, in many real-world scenarios, most of the features are correlated or redundant and thus are not essential and discriminative or even noisy and degrade the clustering process [15, 257]. Moreover, processing high-dimensional data requires significantly increasing time and space. Feature selection aims at finding the most relevant feature subset from high-dimensional feature space based on certain evaluation criteria [15–17]. Thus, feature selection helps in improving the statistical model structure and overcoming several issues that may be caused by the high dimensionality of the feature space such as over-fitting, low efficiency, and poor performance [18–20]. Compared to continuous data, the larger number of features and sparsity nature of discrete data make the feature selection task more critical. Typical examples that involve thousands of discrete features include gene microbiology data [258, 259], social media, and web pages that are generally represented as frequencies of the corresponding set of keywords [260, 261], and visual items (images and videos) that incorporate an extensive number of keypoints [262, 263]. The widely used feature selection methods can be categorized in regard of the way of utilizing label information as supervised algorithms [257, 264], semi-supervised algorithms [265, 266] and unsupervised algorithms [16, 267–269]. In real-world applications, high-dimensional unlabeled data are rapidly accumulated, and obtaining labeled data is both expensive and time-consuming [270, 271]. Consequently, developing unsupervised feature selection techniques is absolutely promising yet challenging. The problem of feature selection becomes more demanding in case neither the class labels nor the number of clusters is known, as the selection of both the best features and the optimal number of components have to be performed simultaneously. To date, the model-based unsupervised feature selection approaches for discrete data have not been much investigated in the literature yet.

In this work, we propose an unsupervised learning algorithm considering the feature saliencies, which is able to perform feature selection and clustering for count data simultaneously. The proposed approach is an extension to the approach in [267, 272], which is based on estimating a weight

for each feature, which is a real-valued quantity in $[0, 1]$ with respect to each mixture component. For this estimation, we introduce a novel minorization-maximization (MM) algorithm [273, 274], that generalizes the celebrated EM algorithm [160] and led to simpler derivation, to calculate the maximum likelihood estimates (MLEs) of parameters. Moreover, we optimize the message length of the dataset based on the minimum message length (MML) philosophy [71] that helps to define the number of relevant components. Given the significance of selecting an accurate model that approximates the distribution of the data, we adopted the marginal density called Generalized Dirichlet Multinomial (GDM) [10], that is the composition of the generalized Dirichlet distribution [240] and the multinomial. Indeed, GDM distribution is more versatile than other widely-used models for count data, including the multinomial and Dirichlet Compound Multinomial (DCM) [9], and it has proved its flexibility and efficiency in several machine learning problems [10, 100, 175]. For modeling high dimensional data, considering the conditional independence assumption among features is a common practice by researchers [267, 275]. Thus, we take advantage of the GDM density version proposed in [232] based on replacing of gamma functions by rising polynomials.

Our key contributions are highlighted as follows:

- We propose a probabilistic feature selection approach that considers discrete random variables modeled by a finite mixture of GDM distributions. To the best of our knowledge, the proposed work is the first model-based feature selection algorithm based on hierarchical Bayesian model.

- We derive a minorization-maximization MM algorithm for maximum likelihood estimation of the proposed mixture with feature saliencies where the surrogate function is much simpler than the log-likelihood, and thus the M step can be solved analytically. The proposed work is the first to consider MM in case of incomplete count data and with feature selection.

- We propose an unsupervised learning approach that simultaneously deals with fitting the mixture model to the observed data and selecting the number of components, to avoid the situation that all saliencies take the maximum value and thus allow us to prune the irrelevant feature.

- We validate the proposed model via challenging clustering problems that involve multimedia data with high-dimensional discrete feature spaces.

The remaining of this paper is arranged as follows. We briefly overview the related work for feature selection methods in Section 7.2. Then, we present our proposed formulation in Section 7.3 followed with the developed algorithm for parameters estimation and model selection in Section 7.4. Extensive experiments are conducted and analyzed in Section 7.5. Finally, Section 7.6 concludes this work with directions for future work.

7.2 Related Works

In the era of big data today, the rapid growth of data has led to an exponential scale with respect to dimensionality and sample size. High dimensional data presents several challenges for learning models in terms of training times, algorithmic complexity, and storage space requirements. Consequently, many dimensionality reduction approaches have been heavily studied in the past years, including feature selection and feature extraction (see, for instance, [276–278]). Typically, feature extraction methods, *e.g.,* partial least square (PLS) [279], principal component analysis (PCA) [280] and latent Dirichlet allocation (LDA) [281], transforms a high dimensional feature space into a distinct low dimensional space, which implies an information loss that could have been discriminative [282]. Furthermore, feature extraction approaches suffer from the difficulty of interpretation since the physical meaning of the original features cannot be retrieved [283, 284]. On the other hand, feature selection is the process of selecting a representative subset of the collected features to handle the curse of dimensionality and improve both the efficiency and effectiveness of the learning task dramatically [15, 285, 286].

The feature selection techniques studied in the literature can be grouped into three categories, namely, filter [287, 288], wrapper [289, 290], and embedded/hybrid [284, 291, 292]. Filter methods investigate the feature's properties using a given dataset in order to evaluate its relevance, where the problem of model building is handled independently [114, 259, 288]. They are generally simple, scalable, and timely-efficient. In such methods, a score is assigned to each individual feature according to certain evaluation criteria such as distance, entropy, dependency, and consistency measures [114]. Hence, they only select the top-ranked features, and ignore the rest, without considering the redundancy among features [114, 271]. Furthermore, it is possible that different feature subsets

seem equally good based on the clustering quality measures; thus, an optimal feature subset may not be unique [17, 261]. Wrapper methods, on the other hand, perform cross-evaluation and comparison among subsets of combined features, where the determination of the most efficient subset is achieved with regard to a specific learning algorithm [292]. Using the wrapper feature selection, the classifier is retrained each time a new feature set is generated which has the advantage of the desirable excessive feature space search to improve the classification accuracy by finding a better feature subset, yet, wrappers implicate an egregious computational cost especially for data with a vast number of features [114, 261]. Hybrid methods are filter–wrapper combinations that perform feature selection as a part of the model construction process, and they are used to benefit from the time-efficiency of filters and the clustering quality of wrappers [284, 291]. The majority of feature selection techniques for discrete data that have been discussed in the literature are for supervised learning and widely adopted in text categorization. Given the sparsity nature of such data, wrapper methods are extremely costly and inefficient; thus, filter methods are usually preferred. Popular state-of-the-art filter techniques for text include the TF-IDF measure [294], and filters based on the information theory measures, including mutual information [295], information gain [296], chi-square $\mathcal{X}^2$ statistic [297], and GSS coefficient [298], to name a few. Other examples of widely used filters used in unsupervised feature selection include: feature dependency [299], entropy-based distance [287], and laplacian score [300]. More details, comparisons and discussions can be found in [17, 114, 283, 301].

On the other hand, model-based clustering has been widely acknowledged and successfully applied as a convenient yet flexible formal setting for unsupervised learning. Though several efforts have been made for developing model-based feature selection approaches to improve the accuracy of clustering. Most of the previous works are presented with respect to clustering using Gaussian mixture models (a recent survey of feature selection methods for Gaussian mixture models can be found at [302]). Moreover, there are some model-based feature selection methods that have been proposed for non-Gaussian data [303, 304], and few others focused on discrete data (see for instance; [268, 305, 306]). In [305], the authors proposed a Bayesian approach for integrating feature selection in text document collections clustering based on multinomial mixtures. The authors in

[306] proposed a feature selection approach using multinomial mixtures learned via maximum likelihood estimation without taking the problem of model selection into consideration and applied only for text. Another interesting related work is the feature weighting using maximum a posteriori (MAP) and model selection via stochastic complexity proposed in [268] applied to text clustering and categorization of visual concepts in different image data. This approach combines clustering and feature weighting and is based on a discrete finite mixture of multinomial distributions. Unlike most of the feature selection methods that perform hard feature selection, *i.e.,* a feature is either selected or not, the approach in [268] assigns weights to different features to indicate their significance (feature weighting clustering has been considered in different works to improve the performance, *e.g.,* [267, 272, 303, 307, 308]). The approach proposed in [267] considers the feature weighting in an unsupervised setting as a model-selection-type problem with respect to Gaussian mixture-based clustering. That is, this approach integrates the process of feature saliency estimation into the model selection and estimation process proposed in [1]. Hence, the obtained method is able to select the relevant features and determine the optimal number of clusters simultaneously. Adopting a minimum message length (MML) [41] to select the number of clusters, helps in encouraging the weights (saliencies) of the irrelevant features to go to zero and, thus, avoid the situation where all the saliencies take the maximum possible value. The present work extends this approach to the case of discrete data clustering, taking into account the effective recent extension by Hong et al. [272] that incorporates defining a new feature saliency with respect to each mixture component, rather than having the same feature saliencies for the whole model.

7.3 The Proposed Model

In this section, we outline the proposed feature selection approach based on the GDM mixture model, where we present its intuition and the formal definition. We start by discussing the statistical properties of the considered density, then we define the concept of feature saliency and give the complete model.

7.3.1 An Alternative Representation for Generalized Dirichlet Multinomial (GDM)

The generalized Dirichlet multinomial (GDM) [10] is a composition of the generalized Dirichlet distribution and the multinomial. GDM is a hierarchical Bayesian framework that generalizes the Dirichlet Compound Multinomial (DCM), which has shown to outperform the typically used multinomial (MN) distribution [9, 63]. By considering Generalized Dirichlet (GD) distribution [240], as a prior to the multinomial, several limitations of the DCM distribution in modeling the covariances could be overcome. Precisely, the generalized Dirichlet distribution can release both the negative-correlation and equal-confidence requirements [243]. Moreover, the independence property of GD distribution grants it more flexibility than the Dirichlet, by sampling each entry of the random vector from independent Beta distributions [241]. Indeed, GDM is an efficient and flexible generative model that is more appropriate for count data that are usually characterized by burstiness and overdispersion phenomena [7, 8].

The GD distribution is constructed through breaking the interval $[0, 1]$ into D subintervals of lengths $\rho_1, \ldots, \rho_D$ by choosing D independent beta variates Z_l with parameters α_l and β_l, where the last length $\rho_D = 1 - (\rho_1 + \cdots + \rho_{D-1})$. The GD density of a vector $\boldsymbol{\rho}$ is given by [100]:

$$\mathcal{GD}(\boldsymbol{\rho}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{l=1}^{D} \frac{\Gamma(\alpha_l + \beta_l)}{\Gamma(\alpha_l)\Gamma(\beta_l)} \boldsymbol{\rho}_l^{\alpha_l - 1} \Big(1 - \sum_{l=1}^{l} \boldsymbol{\rho}_l\Big)^{\gamma_l} \tag{7.1}$$

where $\gamma_l = \beta_l - \alpha_{l+1} - \beta_{l+1}$, for $l = 1, \ldots, D-1$, and $\gamma_D = \beta_{D-1} - 1$. As shown in [10], the generalized Dirichlet is a conjugate prior to the multinomial distribution. Thus, the generalized Dirichlet multinomial (GDM) is the marginal distribution of a count vector $\mathbf{X} = (X_1, \ldots, X_{D+1})$ obtained by integrating over m multinomial trail:

$$\mathcal{GDM}(\mathbf{X}|\boldsymbol{\alpha}, \boldsymbol{\beta}) = \binom{m}{X} \prod_{l=1}^{D} \frac{\Gamma(\alpha_l + X_l)}{\Gamma(\alpha_l)} \times \frac{\Gamma(\beta_l + Y_{l+1})}{\Gamma(\beta_l)} \frac{\Gamma(\alpha_l + \beta_l)}{\Gamma(\alpha_l + \beta_l + Y_l)} \tag{7.2}$$

where $Y_l = X_l + X_{l+1} + \cdots + X_{D+1}$.The generalized Dirichlet multinomial distribution include the Dirichlet multinomial as a special case. That is, GDM is reduced to DCM by setting $\beta_l = \alpha_{l+1} + \beta_{l+1}$.

The occurrence of Gamma function in the density function (Eq.7.2) poses an unappealing feature in terms of the complication of evaluating the function and derivatives, which can compromise the performance. Haldane [309] suggested replacing Gamma functions by rising polynomials for an appreciable gain in simplicity. Moreover, re-parameterizing the density parameters in terms of proportion and overdispersion, suggested in [310], has shown its efficiency in implementing Newton's method for maximum likelihood estimation with the beta-binomial distribution [311]. By adopting this re-parameterization concept in the case of GDM, such that:

$$\theta_l = \frac{1}{\alpha_l + \beta_l}, \quad \pi_l = \frac{\alpha_l}{\alpha_l + \beta_l}, \quad l = 1, \ldots, D,$$

and using the fact that $X_l + Y_{l+1} = Y_l$, Zhou and Lange [232] re-expressed the GDM density as D-independent beta-binomial:

$$\mathcal{P}(\mathbf{X}|\boldsymbol{\theta}, \boldsymbol{\pi}) = \binom{m}{X} \prod_{l=1}^{D} \frac{\pi_l \ldots [\pi_l + (X_l)\theta_l] \times (1-\pi_l) \ldots [1 - \pi_l + (Y_{l+1} - 1)\theta_l]}{1 \ldots [1 + (Y_l)\theta_l]} \tag{7.3}$$

This version of GDM density has been successfully used in estimating the maximum likelihood [232], where the estimation of the parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_D)$ and $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_D)$ is reduced to the case of D independent beta-binomial estimation problems.

7.3.2 Mixture Model with Feature Saliency

Let $\mathcal{X} = \{X_1, \ldots, X_N\}$ be a dataset of N objects, and each $X_i = (X_{i1}, \ldots, X_{iD+1})$, is a $D+1$-dimensional count vector. We assume that the features are conditionally independent given the hidden component label, that is:

$$P(\mathcal{X}|\Theta_M) = \prod_{i=1}^{N} \sum_{j=1}^{M} p_j \prod_{l=1}^{D} \mathcal{P}(X_{il}|\theta_{jl}, \pi_{jl}) \tag{7.4}$$

which is an M-component mixture model, and $\Theta_M = \{\{\pi_j\}, \{\theta_j\}, (p_1, \ldots, p_M)\}$ is the set of parameters defining the mixture model, where X_{il} denotes the lth feature and θ_{jl}, and π_{jl} denote the parameter corresponding to the lth feature in the jth component.

Taking into account the high-dimensionality and sparsity nature of count data, as well as the fact

that all features are not equally important for clustering, feature selection is a well-known approach to improve the accuracy of the model [283]. For defining the feature relevancy in our model, we adopt the concept of feature saliency, which has shown to be appropriate for unsupervised learning (*e.g.,* [267, 268, 272]). Following the practice in [272], we define a set of binary parameters to represent the feature relevance $\Phi = \{\phi_1, \ldots, \phi_M\}$ such that $\phi_{jl} = 1$ if the lth feature is relevant to the jth component and $\phi_{jl} = 0$ otherwise. Typically, a given feature is assumed to be independent of the class labels following a common density across classes [305, 312]. Indeed, the common density reflects our prior knowledge about the distribution of the nonsalient features. Let $\Xi = \{\Lambda, \mu\}$ where $\Lambda = (\lambda_1, \ldots, \lambda_D)$, and $\mu = (\mu_1, \ldots, \mu_D)$ are the parameters of the generalized Dirichlet multinomial GDM that is defined as a common density across classes to explain non-salient features, then $P(\mathcal{X})$ can be approximated as:

$$P(\mathcal{X}|\Theta_M, \Phi, \Xi) = \prod_{i=1}^{N} \sum_{j=1}^{M} p_j \prod_{l=1}^{D} [\mathcal{P}(X_{il}|\theta_{jl}, \pi_{jl})]^{\phi_{jl}} \times [\mathcal{P}(X_{il}|\lambda_l, \mu_l)]^{1-\phi_{jl}} \tag{7.5}$$

which is reduced to Eq.7.4 when $\phi_{jl} = 1$ for each $j = 1, \ldots, M$ and $l = 1, \ldots, D$, *i.e.*, all features are relevant with respect to all the components. Next, we introduce the component-based feature saliency as the probability that lth feature is relevant to jth component $\rho_{jl} = p(\phi_{jl} = 1)$. Thus, Φ is a set of missing variables following a multiple Bernoulli distribution $p(\Phi) = \prod_{l=1}^{D} \rho_{jl}^{\phi_{jl}}(1 - \rho_{jl})^{1-\phi_{jl}}$, we can write the mixture density as:

$$P(\mathcal{X}|\Theta_M, \Phi, \Xi) = \prod_{i=1}^{N} \sum_{j=1}^{M} p_j \prod_{l=1}^{D} [\rho_{jl}\mathcal{P}(X_{il}|\theta_{jl}, \pi_{jl})]^{\phi_{jl}} \times [(1 - \rho_{jl})\mathcal{P}(X_{il}|\lambda_l, \mu_l)]^{1-\phi_{jl}} \tag{7.6}$$

By marginalizing the previous equation over Φ, and considering that ϕ_{jl} is binary, we obtain our mixture model with feature saliency/weighting, as:

$$P(\mathcal{X}|\Theta) = \prod_{i=1}^{N} \sum_{j=1}^{M} p_j \prod_{l=1}^{D} \Big[\rho_{jl}\mathcal{P}(X_{il}|\theta_{jl}, \pi_{jl}) + (1 - \rho_{jl})\mathcal{P}(X_{il}|\lambda_l, \mu_l)\Big] \tag{7.7}$$

where $\Theta = \{\{\pi_{jl}\}, \{\theta_{jl}\}, \{p_j\}, \{\boldsymbol{\rho}_{jl}\}, \{\lambda_l\}, \{\mu_l\}\}$ is the set of all the parameters of the model.

The generative interpretation of our proposed model is as follows. The model starts with selecting the component label j by sampling from the GDM distribution with mixing parameters $p_1, \ldots, p_M$, then each feature $X_{i1}, \ldots, X_{iD}$ has a binomial experiment with two possible outcomes: we use the jth mixture component to generate the lth feature, or we use the common component to generate it. That is, the probability of feature relevancy depends on the mixture component. Moreover, the proposed model has the advantage of avoiding data overfitting as the added parameters grants it additional degrees of freedom.

7.4 Model Learning

7.4.1 Parameters Estimation

Define $\mathcal{Z} = \{Z_1, \ldots, Z_N\}$, $Z_i = (z_{i1}, \ldots, z_{iM})$ as the hidden class labels with $z_{ij} = 1$ if $\mathbf{X}$ belongs to class j and 0 otherwise. Then, letting $(\mathcal{X}, \mathcal{Z})$ be the complete dataset, the data log-likelihood is given by:

$$\log P(\mathcal{X}, \mathcal{Z}|\Theta) = \sum_{i=1}^{N}\sum_{j=1}^{M} z_{ij} \log \left[p_j \prod_{l=1}^{D} \left[\rho_{jl}\mathcal{P}(X_{il}|\theta_{jl}, \pi_{jl}) + (1-\rho_{jl})\mathcal{P}(X_{il}|\lambda_l, \mu_l) \right] \right] \tag{7.8}$$

Model fitting is one of the objectives of mixture model-based clustering achieved by inferring Θ from a given dataset $\mathcal{X}$. Then, the clustering is obtained by assigning each data point to a different component, *i.e.,* the component with the highest posterior probability of membership. Thus, we need to find the component index corresponding to the highest value of the posterior probability value p_{ij}, which is the probability that $\mathbf{X}_i$ is generated from the jth component of the mixture, as:

$$\begin{aligned} p_{ij} &= P(\mathbf{X}_i, z_{ij} = 1, z_{ik,k\neq i} = 0|p, \rho, \theta, \pi, \lambda, \mu) \\ &= p_j \prod_{l=1}^{D} \left[\rho_{jl}\mathcal{P}(X_{il}|\theta_{jl}, \pi_{jl}) + (1-\rho_{jl})\mathcal{P}(X_{il}|\lambda_l, \mu_l) \right] \end{aligned} \tag{7.9}$$

The estimation of the model parameters is performed by maximizing Eq.(7.8) given a dataset $\mathcal{X}$. The widely used as the most effective approach for maximizing the log-likelihood is the expectation-maximization (EM) algorithm [160], which generates a sequence of models with non-decreasing

log-likelihood on the data. Indeed, EM is a special case of a more general MM principle initially introduced by Ortega and Rheinboldt [273]. Several merits can be obtained by considering the MM principle mainly for attacking optimization problems and computational balance. MM algorithms have attractive features, including that they are numerically stable, can be coded easily, and its convergence can be accelerated [232, 274]. The MM principle has been considered in machine learning and statistical estimation in two different versions: in minimization problems, where the first M stands for majorize and the second M for minimize [313], and in maximization problems, where the first M stands for minorize and the second M for maximize [232, 314].

Here, we utilize the MM principle for a maximization problem, precisely to calculate the maximum likelihood estimates (MLEs) of the proposed model parameters and the posterior modes for the analysis of incomplete count data. MM algorithm does not depend on the choice of the initial values and can avoid the complication in calculating the Hessian matrix. Let $f(\theta)$ be the objective function we seek to maximize, which is in our case the complete data log-likelihood given by Eq.(7.8), the minorization step is performed by construction of the surrogate function $g(\theta|\theta^n)$, which is defined by the following two properties (see Appendix 1):

$$f(\theta^n) = g(\theta^n|\theta^n), \tag{7.10}$$

$$f(\theta) \geq g(\theta|\theta^n), \qquad \theta \neq \theta^n. \tag{7.11}$$

In the second M, we maximize the surrogate function $g(\theta|\theta^n)$ instead of the complete log-likelihood function $f(\theta)$. By treating $\mathcal{Z}$ and Φ as hidden variables, the maximization step includes running the EM algorithm for evaluating the posterior distribution over latent variables and estimate

the parameters. In E-step, we compute the following quantities [272]:

$$a_{ijl} = P(\phi_{jl} = 1, X_{il}|Z_i = j) = \rho_{jl}\mathcal{P}(X_{il}|\theta_{jl}, \pi_{jl}), \tag{7.12}$$

$$b_{ijl} = P(\phi_{jl} = 0, X_{il}|Z_i = j) = (1 - \rho_{jl})\mathcal{P}(X_{il}|\lambda_l, \mu_l), \tag{7.13}$$

$$c_{ijl} = P(X_{il}|Z_i = j) = a_{ijl} + b_{ijl}, \tag{7.14}$$

$$\omega_{ij} = P(Z_i = j|X_i) = \frac{p_j \prod_l c_{ijl}}{\sum_j p_j \prod_l c_{ijl}}, \tag{7.15}$$

$$\upsilon_{ijl} = P(Z_i = j, \phi_{jl} = 1|X_i) = \frac{a_{ijl}}{c_{ijl}}\omega_{ij}, \tag{7.16}$$

$$\nu_{ijl} = P(Z_i = j, \phi_{jl} = 0|X_i) = \omega_{ij} - \upsilon_{ijl} \tag{7.17}$$

Then, in M-step the parameters estimation is performed according to:

$$\hat{p_j} = \frac{1}{N}\sum_{i=1}^{N}\omega_{ij}, \tag{7.18}$$

$$\hat{\rho_{jl}} = \frac{\sum_{i=1}^{N}\upsilon_{ijl}}{\sum_{i=1}^{N}\omega_{ij}} \tag{7.19}$$

Such that we calculate feature saliencies specific to each model component. Equating the partial derivative of the surrogate function (see Eq.(7.33) in Appendix 1) with respect to each parameter to 0 yields the following updates:

$$\hat{\pi_{jl}} = \sum_i \upsilon_{ijl}\left(\sum_k \frac{r_{lk}\pi_{jl}}{\pi_{jl} + k\theta_{jl}}\right) / \left(\sum_k \left[\frac{r_{lk}\pi_{jl}}{\pi_{jl} + k\theta_{jl}} + \frac{s_{l+1k}(1 - \theta_{jl})}{1 - \pi_{jl} + k\theta_{jl}}\right]\right) \tag{7.20}$$

$$\hat{\theta_{jl}} = \sum_i \upsilon_{ijl}\left(\sum_k \left[\frac{r_{lk}k\theta_{jl}}{\pi_{jl} + k\theta_{jl}} + \frac{s_{l+1k}k\theta_{jl}}{1 - \pi_{jl} + k\theta_{jl}}\right]\right) / \left(\sum_k \frac{s_{lk}}{1 + k\theta_{jl}}\right) \tag{7.21}$$

$$\hat{\mu_l} = \sum_i (\sum_j \nu_{ijl})\left(\sum_k \frac{r_{lk}\mu_l}{\mu_l + k\lambda_l}\right) / \left(\sum_k \left[\frac{r_{lk}\mu_l}{\mu_l + k\lambda_l} + \frac{s_{l+1k}(1 - \lambda_l)}{1 - \mu_l + k\lambda_l}\right]\right) \tag{7.22}$$

$$\hat{\lambda}_l = \sum_i (\sum_j \nu_{ijl}) \left(\sum_k \left[\frac{r_{lk} k \lambda_l}{\mu_l + k\lambda_l} + \frac{s_{l+1k} k \lambda_l}{1 - \mu_l + k\lambda_l} \right] \right) / \left(\sum_k \frac{s_{lk}}{1 + k\lambda_l} \right) \tag{7.23}$$

In these equations, the variable υ_{ijl} measures the importance of the ith pattern to the jth component in case of using the lth feature. similar interpretation can be applied to the common distribution, *i.e.,* same relationship exists between $\sum_i \upsilon_{ijl}$ and λ_l, μ_l. Moreover, the estimation of ρ_{jl} is proportional to $\sum_i \upsilon_{ijl}$ explained by how likely it is that ϕ_{jl} equals to one with respect to a mixture component j. To avoid underflow in likelihood computations, all probabilities are represented as logarithms. Moreover, to avoid overflow without losing precision, we followed the practice in [12] where class responsibilities ω_{ij} are computed as:

$$\omega_{ij} = \frac{\exp\left(\log p_j + \sum_{l=1}^{D} \log(c_{ijl}) - \varepsilon \right)}{\sum_j \exp\left(\log p_j + \sum_{l=1}^{D} \log(c_{ijl}) - \varepsilon \right)} \tag{7.24}$$

where $\varepsilon = \max_j \{ \log p_j + \sum_{l=1}^{D} \log(c_{ijl}) \} - 100$.

7.4.2 Model Selection

The minimum message length (MML) criterion for a mixture of distributions is [40, 41]:

$$\hat{\Theta}_{MML} = \underset{\Theta}{\operatorname{argmin}} \left\{ -\log(P(\Theta)) - \log(P(\mathcal{X}|\Theta)) + \frac{1}{2} \log |F(\Theta)| + \frac{N_p}{2} \left(1 + \log \frac{1}{12} \right) \right\} \tag{7.25}$$

where Θ is the set of the model parameters, N_p is the dimension of Θ, $P(\mathcal{X}|\Theta)$ is the complete data likelihood, and $F(\Theta) = -E[D_\theta^2 \log P(\mathcal{X}|\Theta)]$ is the expected Fisher Information Matrix (FIM) which is the negative expected value of the Hessian of the log-likelihood. In our case, the FIM is very difficult to obtain analytically. Thus, we adopted the approach [1, 267] by approximating the information matrix of complete data log-likelihood as a block diagonal matrix of size $(M + D + MDR + DS)$, where R and S are the number of θ_{jl}, π_{jl} and λ_l, μ_l parameters, respectively. Furthermore, given the lack of knowledge about mixture parameters, we adopt the standard non-informative Jeffreys' prior [189], which are proportional to the square root of the determinant of the corresponding information matrices (see [1, 267] for details). The MML criterion for our model

consists of minimizing, with respect to Θ, the following cost function:

$$MessLength = -\log P(\mathcal{X}|\Theta) + \frac{M+D}{2}\log N + \frac{R}{2}\sum_{l=1}^{D}\sum_{j=1}^{M}\log(Np_j\rho_{jl}) + \frac{S}{2}\sum_{l=1}^{D}\sum_{j=1}^{M}\log(N(1-\rho_{jl})), \tag{7.26}$$

where the number of parameters $R = S = 2$. From a parameter estimation point of view, minimizing the message length in Eq.(7.26) is equivalent to maximizing the complete data log-likelihood. Using a Dirichlet-type priors on p_j's and ρ_{jl}'s, the M-step (7.18) and (7.19) are replaced by:

$$\hat{p_j} = \frac{\max(\sum_i \omega_{ij} \frac{RD}{2}, 0)}{\sum_j \max(\sum_i \omega_{ij} - \frac{RD}{2}, 0)}, \tag{7.27}$$

$$\hat{\rho_{jl}} = \frac{\max(\sum_i \upsilon_{ijl} - \frac{MR}{2}, 0)}{\max(\sum_i \upsilon_{ijl} - \frac{MR}{0}, 0) + \max(\sum_i \nu_{ijl} - \frac{S}{2}, 0)} \tag{7.28}$$

These forms for updating the mixing weights and feature saliencies have a pruning behavior such that they force some component with a very small weight p_j to be pruned, and some of the feature saliency ρ_{jl} to go to zero, so the feature is no longer salient, or one so it can be dropped.

7.4.3 The Complete Unsupervised Feature Saliency Algorithm

The proposed unsupervised algorithm for simultaneous feature selection and clustering is summarized in Algorithm 7. First, a lower bound on the number of components will be specified M_{min} and the algorithm will be initialized with a large number of components[1] M_{max} making it less dependent of the initial values of the parameters. The mixing weight $\{p_j\}^{(0)}$ will be initialized using the K-means algorithm to assign each data point initially to one of the $1, \ldots, M_{max}$ components, then, the mixture parameters $\{\pi_{jl}\}^{(0)}$, $\{\theta_{jl}\}^{(0)}$, and common distribution parameters $\{\lambda_l\}^{(0)}$, $\{\mu_l\}^{(0)}$ initialization will be given randomly while feature saliency $\{\boldsymbol{\rho}_{jl}\}^{(0)}$ has a fixed initial value of 0.5. As starting with a large value of M may lead to several empty components, there will be no need to estimate and transmit their parameters. Thus, we adopt the component-wise EM procedure (CEM) [187], as proposed in [1, 267], instead of the EM in the maximization step. That

[1] In our experiments, the values for M_{min} and M_{max} have been set to 2 and 50, respectively.

is, while the number of non-zero components $M^+ \geq M_{min}$, the E-step of the algorithm will run where we evaluate the posterior probabilities for each component and estimate the mixture proportion for that component. Then, if the component becomes irrelevant, with $p_j^{(t+1)} = 0$, it will be annihilated; otherwise, their parameter updates should be performed, and the MML criterion is re-evaluated for non-zero components only. The convergence will be achieved when the change in the message length LEN, or alternatively in the log-likelihood function, becomes insignificant. Each computation of LEN requires $\mathcal{O}(ND(M+1))$ time.

Algorithm 7: The unsupervised feature saliency algorithm using GDM.

Output: Optimal number of components M^*, best mixture parameters $\{\pi_{jl}\}, \{\theta_{jl}\}$, best common density parameters $\{\lambda_l\}, \{\mu_l\}$, and feature saliencies $\{\boldsymbol{\rho}_{jl}\}$.

Input: Dataset $\mathcal{X} = \{X_1, \ldots, X_N\}$, M_{min}, M_{max}

Initialize: $\Theta^{(0)} = \{\{\pi_{jl}\}, \{\theta_{jl}\}, \{p_j\}, \{\boldsymbol{\rho}_{jl}\}, \{\lambda_l\}, \{\mu_l\}\}$
for $j = 1, \ldots, M_{max}, \quad l = 1, \ldots, D$. $t \leftarrow 0, \quad M^+ = M_{max}, \quad LEN_{min} = +\infty$. ;

while $M^+ \geq M_{min}$ **do**
  repeat till finding the local minimum;
  for $j = 1 \rightarrow M^+$ **do**
    Perform E-step according to Eqs.(7.12) to (7.17) Update the mixing proportion and feature salincies according to Eqs.(7.27) and (7.28) ;
    Update the common distribution parameters according to Eqs.(7.22) and (7.23);
    if $p_j^{(t+1)} > 0$ **then**
      Update the mixture densities parameters according to Eqs.(7.20) and (7.21);
    else
      $M^+ = M^+ - 1$;
    end
  end
  Compute optimal length for the non-zero components $LEN^{(t+1)}$ according to Eq.(7.26);
  if $LEN^{(t+1)} < LEN_{min}$ **then**
    Record the current model parameters ;
    set $LEN_{min} = LEN^{(t+1)}$
  end
  $t \leftarrow t + 1$
end

7.5 Experimental Results

The objective of our experiments is to validate the efficiency of the proposed framework in clustering high-dimensional count data. We have compared the performance of the proposed GDM mixture model based on MM updates with and without feature selection to the previously proposed mixture of GDM based on the maximum likelihood estimation approach [10]. Moreover, we compared with the performance of two widely considered generative models for count data, namely, the multinomial mixture model (MM) [315], and the mixture of Dirichlet Compound Multinomial (DCM) [9]. All our experiments have been conducted using optimized R2017a MATLAB codes on an Intel(R) Core(TM) i7-6700 Processor PC with the Windows 7 Enterprise Service Pack 1 operating system with a 16 GB main memory. In the following sections and through a set of real-world applications that involve high-dimensional textual and visual data, we will demonstrate the importance of feature selection in improving clustering results.

7.5.1 Hate Speech and Offensive Language Detection on Twitter

Social networks nowadays are suffering an increase in the offensive, abusive, or hateful language that exposes sexism, racism, and other types of aggressive and cyberbullying behavior. Thus, there has been great interest, by both academy and industry, in proposing systems to detect and block abusive behavior automatically. Several challenges involve abusive content detecting, including the ambiguity in defining what qualifies as an abuse, which makes extracting ground truth a challenging task. In addition, the accounts carrying such content are usually controlled by humans using a massive amount of vocabulary, with a misspelling, abbreviations, and extra letters or punctuation that can help them to express negative emotions or sarcasm. Another challenge that this behavior is relatively uncommon where only a few examples can be found from a random collection resulting in an issue of imbalanced classes. To evaluate our proposed framework in detecting hate speech and offensive language, we used a recent dataset by Davidson et al. [316] that contains $25k$ collected tweets in three categories distinguishing between hate speech (language that is used to express hatred towards a targeted group based on characteristics like race, ethnicity, religion, gender, and sexual orientation language or to insult the members of the group), offensive but not hate speech,

Table 7.1: Clustering results (%) for Twitter dataset over 20 random runs.

Model	Precision	Recall	F-measure	BE_{micro}	BE_{macro}
MM	37.57	36.73	37.64	37.57	58.79
DCM	63.32	59.66	61.44	63.32	78.87
GDM	59.99	74.69	66.54	59.99	83.79
GDM-MM	66.09	77.50	71.34	66.09	82.41
GDM-FS	69.47	83.79	75.96	69.47	84.39

or neither offensive nor hate speech. Each tweet was assigned to one of the classes based on the majority decision by three or more people. The majority of the tweets in the considered dataset are categorized as an offensive language (76.8%), where only (5.7%) contains hate speech, and the reminder (16.6%) considered to be normal (neither offensive nor hate speech) with a total of $67,094$ unique words.

As our data is unbalanced and multi-labeled, it is crucial to take both precision and recall into consideration to ensure a fair measurement of performance. Thus, we combine both measures by computing the F score, and break-even point where precision and recall are equal averaged at the micro or macro level, as [9]:

$$BE_{micro} = \frac{1}{N}\sum_{j=1}^{M} N_j \frac{TP_j}{TP_j + FP_j} \tag{7.29}$$

$$BE_{macro} = \frac{1}{M}\sum_{j=1}^{M} \frac{TP_j}{TP_j + FP_j} \tag{7.30}$$

where M is the number of classes, N is the total number of documents, N_j is the number of documents in class j, TP_j and FP_j are the number of true and false positives per class, respectively.

Table 7.1 presents the performance of each of the tested models in clustering the tweets into one of the three labels *i.e.,* hate speech, offensive language, or neither. First, based on the break-point at the macro level, we can notice that the mixture of GDM has achieved an overall better performance compared to the multinomial and DCM mixtures. However, as shown in the first confusion matrix in Fig. 7.1, around 30% of hate speech is misclassified: the precision and recall scores for the hate class are 0.31 and 0.69, respectively. Tweets with the highest predicted probabilities of being

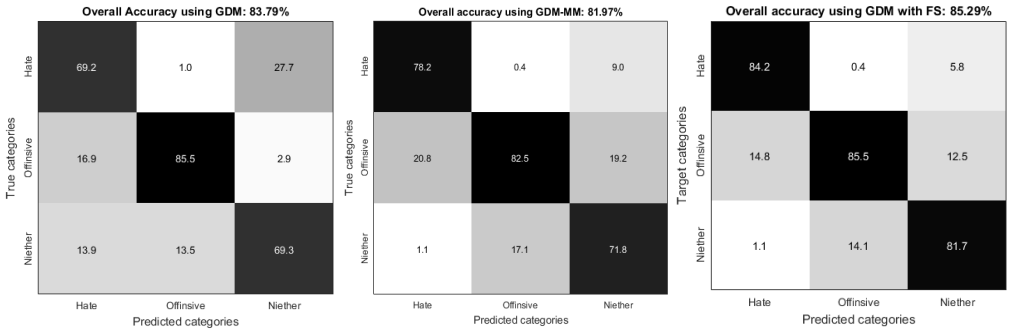


Figure 7.1: Confusion matrices for detecting hate speech and offensive language using different approaches.

hate speech tend to contain multiple racial or homophobic slurs. Furthermore, tweets are classified as hate when people use hate speech to respond to other hate speakers, for instance, to criticize someone else racism. The tweets that do not contain any hate speech or offensive language were also misclassified, where around 28% have been predicted as hate tweets. Compared to Newton's method, the MM solution has shown to improve the performance at class level (for two out of the three classes) where the micro averaged BE has been improved from 59.99% to 66.09%. This can be explained by the numerical stability of the MM algorithm and by its scalability to high-dimensional data. We can see from the second confusion matrix in Fig. 7.1 that tweets that contain hate speech (sexism, racism, and homophobia) are easier to be detected, resulting in improving the hate class recall from 0.69 to 0.78. Furthermore, from Table 7.1 we observe that the proposed GDM mixture with feature saliency is significantly superior to the same model with both EM and MM algorithms, in terms of the clustering accuracy. Given that our proposed framework considers features relevance to each component of the mixture model, the keywords that characterize each of the hate speech or abusive/offensive text will get the proper relevance with respect to each component and thus distinguish between individual clusters. As a result, we can see from the last confusion matrix in Fig. 7.1 the improved separation of overlapping mixtures, *i.e.,* lower number of tweets in each class were misclassified.

Figure 7.2: Sample images from the considered classes in PublicFig+LFW dataset.

7.5.2 Face Identification

Face recognition and verification problems have attracted the scholars' attention for more than two decades due to their importance in many real-world applications, including video surveillance, criminal investigation, and human behavior analysis. Thus, automated face tagging and recognition forms have been integrated into several consumer platforms such as Google Picasa, Microsoft Photo Gallery, and social network sites such as Facebook. Indeed, the amount of photo and video content has grown significantly, making their organization and searching for photos a considerably challenging task. Moreover, face recognition is being increasingly integrated into more specialized devices such as smartphones, by Apple and Google, in advanced features such as Android face unlocks authentification, and Apple iPhoto face detectors. One of the large-scale datasets that have been recently created for face verification is the PubFig+LFW dataset [317] that combine the Public Figures (PubFig) [318] and the Labeled Faces in the Wild (LFW) [319] datasets of real-world images of public figures (celebrities and politicians) acquired from the web. This dataset has 83 individuals where all the faces from each individual were pre-divided into two-thirds training faces and one third testing faces. We considered a subset of the PubFig+LFW dataset that consists of the faces of 10 celebrities. Sample images from the considered subset are illustrated in Fig.7.2.

The face identification results of our proposed framework, as well as a comparison with the other tested models, are presented in Table 7.2. For most of the classes, the proposed GDM mixture with feature saliency performs slightly better than both GDM with Newton's method and with MM solution, where they themselves outperform the multinomial and DCM mixtures. From the

Table 7.2: Average accuracy (%) per-class for face identification over 20 random runs.

Celebrity	MM	DCM	GDM	GDM-MM	GDM-FS
Angelina Jolie	67.79	46.15	50.77	50.77	83.08
Barack Obama	72.62	57.76	60.71	65.48	78.57
Brad Pitt	63.04	86.96	76.09	76.09	76.09
Cristiano Ronaldo	70.91	85.45	94.55	94.55	85.45
George Clooney	57.75	64.79	83.10	84.51	71.83
Jennifer Aniston	85.75	80.88	83.82	86.76	92.65
Jennifer Lopez	53.29	53.85	69.23	69.23	71.83
Julia Roberts	57.14	66.67	69.05	69.05	69.05
Shahrukh Khan	72.92	85.42	85.42	85.42	83.33
Shakira	61.67	86.67	65.00	71.67	68.33
Overall Accuracy	66.99	71.47	73.40	75.16	78.37

table, we can notice that overall clustering accuracy has been improved, using the MM solution with and without considering the feature saliency to 78% and 75%, respectively. This improvement is statistically significant, as shown by a student t-test (p-values are 0.12 and 0.09 for the different models). Another improvement gain is the simplicity of MM updates that overcomes its slower convergence where it often requires several hundred iterations despite saving overall computational work comparing to the other algorithm (mainly in computing the gamma/digamma/trigamma functions). Moreover, by running our algorithm 20 times, the average number of clusters found was equal to 10.36 ± 0.21. This suggests that feature selection may assist the determination of the optimal number of clusters considering that feature selection and clustering are actually strongly related and can be reinforced by each other.

7.5.3 Race Recognition

Automatic estimation of demographic information including age, gender, and race of a person from his face image could enhance the performance of face recognition, and it has many potential significant applications in different fields ranging from forensics to social media. Indeed, several significant pieces of evidence support the fact that information from various visual cues is utilized to recognize faces. Furthermore, recognizing race has significant implications for the field of public health, where information on race and/or ethnicity has been essential in understanding the health

issues affecting an individual population. We used the UTKFace dataset [320] that consists of $20k+$ single face images cropped to 227×227 around the face center. This dataset is publicly available with a good variety of ages, gender, and ethnicity. The race groups are White (53.9%), Black (4.1%), Asian (15.8%), Indian (15.1%), and Others like Hispanic, Latino, and Middle Eastern (11.1%). Some sample face images from five race groups across the different genders and age labels are shown in Fig. 7.3.

Figure 7.3: Sample images from UTK faces dataset.

As can be seen in Table 7.3, our approach using GDM with feature saliency was able to achieve a high clustering accuracy of 89%, above the 84%, and 85% achieved by GDM with Newton's approach and MM solution without FS, respectively. The improvement obtained using the proposed framework with feature saliency over the other two models is statistically significant according to the student t-test (*i.e.,* p-values are 0.31 and 0.11 for the difference between the proposed model and Newton's method and MM without FS, respectively). In addition, the interclass confusion matrix obtained using our proposed framework is shown in Fig. 7.4, where one may notice that the most of the confusion takes place between *Asian* and *White* as well as between *Indian* and *White*. This is due to the fact that this dataset is clearly intensely imbalanced, *i.e.,* more than half of the images are in the White group making, some features of the other groups hard to be defined and probably confused with the ones from the White group.

Furthermore, we conducted another experiment where we supposed that both the class label information and the number of clusters were unavailable. Based on our experiments, the average number of clusters found over 20 runs using different random initial values is 5.11 ± 0.06. Both the accuracy and selected number of clusters confirmed the efficiency of the proposed model in race

Table 7.3: Average clustering results (%) for race recognition over 20 random runs.

Model	Accuracy (%)
MM	73.96 ± 0.04
DCM	77.01 ± 0.12
GDM	84.50 ± 0.07
GDM-MM	85.80 ± 0.35
GDM-FS	89.01 ± 0.22

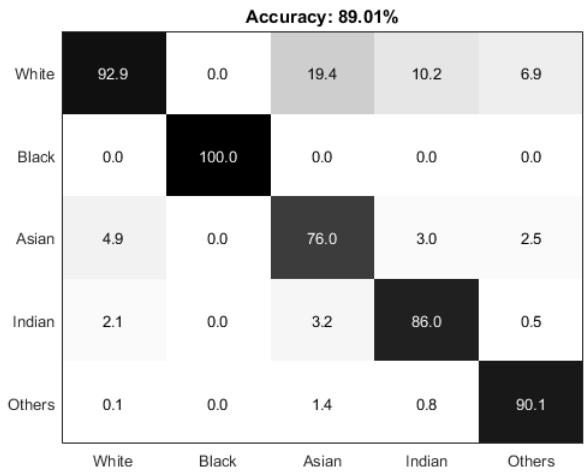


Figure 7.4: Confusion matrix obtained by GDM with FS for race recognition in the UTK face dataset.

recognition based on facial images. It is noteworthy that despite the fact that by initializing a large number of components the algorithm is less sensitive to the local maxima, the speed of convergence will likely be similar for all starting values far from the maximum as MM algorithms usually tend to get close to the answer quickly and then slow down.

7.5.4 Age Estimation

Automatic age estimation is a challenging problem given the fact that persons belonging to the same age group can be extremely different in their facial appearances, which are affected by several intrinsic and extrinsic factors. It has recently received considerably increasing attention due to its significance in many applications such as access control, for instance, where an automatic age estimation system can prevent minors from purchasing alcohol or cigarette from vending machines.

Rather than estimating the precise age, we followed the practice of classifying people into specified age groups. We implement the age estimation using our proposed model in face datasets. The first is MORPH [321] that contain around $55k$ unique images of more than $13,000$ adults with ages range from 16 to 77. The age groups in this dataset include the young people (age<30), the middle-aged people (30<=age<50) and the aged (age>=50) people account for 41%, 53%, and 6% of the dataset respectively. Following the strategy in [322, 323] we selected a subset of the large-scale MORPH dataset with around $20k$ images, which cover all the age groups. Although MORPH is the largest and most commonly used dataset for age estimation, it consists of adult faces only. Thus, we adopted another dataset with a longer age span, namely, the UTKFace dataset [320]. We have created five relatively balanced age groups based on the age labels annotated in this dataset as: infants (age<=2), children (2<age<=12), youth (12>age<=25), adults (25<age<=50), and seniors (age>50).

Table 7.4: Average clustering results (%) for age estimation over 20 random runs.

Model	MORPH	UTK
MM	70.52	66.56
DCM	71.17	69.53
GDM	74.45	72.30
GDM-MM	76.39	75.31
GDM-FS	77.51	78.24

Table 7.4 presents the average accuracy for age estimation in the two tested datasets using the different models. All experiments were performed 20 times in the different models to obtain the mean estimation accuracy for the age group as a performance measure. As shown in the table, our approach obtains the best results among the tested methods in both datasets. For instance, GDM with FS obtains the highest accuracy in the UTK face dataset, where the accuracy achieved is about 2.93%, 5.94%, 8.71%, and 11.68% higher than those of GDM-MM, GDM-EM, DCM, and MM, respectively. This improvement is statistically significant, according to the student t-test (p-value is between 0.18 and 0.23 for the different models).

Furthermore, in a large scale dataset, such as MORPH, although the improvement obtained by

MM solution over the EM is not statistically significant, the EM algorithm involves solving a non-trivial maximization problem in the M step. Thus, compromising the relatively faster convergence with the simplicity of MM updates that converge slowly is crucial. In addition, the performance of our mixture selection model was evaluated on the two challenging datasets, and the average numbers of clusters found over the 20 runs were 3.40 ± 0.03 and 4.98 ± 0.28 for MORPH and UTK face datasets, respectively. Both clustering accuracy and the selected optimal number of components confirmed that our proposed framework is capable of providing promising results in modeling high-dimensional count data.

7.6 Conclusion

In this paper, we have presented a novel statistical framework that considers various critical issues in mixture modeling, including high-dimensional features space, choice of the probability density functions, estimation of the mixture parameters, and automatic determination of the number of mixture components. The proposed framework has several advantages ranging from avoiding multiple runs for each candidate model by initializing a large number of components and annihilating the irrelevant ones, to the simplicity of MM algorithm that avoids the burdensome computation of inversion of large matrices *i.e.,* the Hessian or Fisher information matrix. Furthermore, our model differs from most of the existing works on discrete data feature selection, which are mainly carried out in a supervised learning setting, or considering the generic multinomial model. The developed framework in this paper is an unsupervised component-based feature selection approach, based on a mixture of GDM densities, that determines the optimal number of components and the class assignment simultaneously. Our experiments have shown that assigning a weight to each feature that shows its relevance to each class achieves better performance than assuming that all features are equally important. The efficiency of the proposed framework was demonstrated on textual, and visual datasets represented as high-dimensional count data.

Appendix 1: Surrogate Function Construction

The complete data log-likelihood of the proposed model (Eq. 7.7) is given by:

$$\begin{aligned}\mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{M} p_j \Bigg[& \sum_{l=1}^{D} \log(\rho_{jl}) + \log(\pi_{jl} \ldots [\pi_{jl} + (X_{il}-1)\theta_{jl}]) \\ & + \log\Big((1-\pi_{jl}) \ldots [1-\pi_{jl} + (Y_{il+1}-1)\theta_{jl}]\Big) \\ & - \log(1 \ldots [1 + (Y_{il}-1)\theta_{jl}]) + \log(1-\rho_{jl}) \\ & + \log(\mu_l \ldots [\mu_l + (X_{il}-1)\lambda_l]) \\ & + \log\Big((1-\mu_l) \ldots [1-\mu_l + (Y_{il+1}-1)\lambda_l]\Big) \\ & - \log(1 \ldots [1 + (Y_{il}-1)\lambda_l]) \Bigg] \end{aligned} \tag{7.31}$$

To construct an MM algorithm, we need to minorize terms such as $\log(\pi_{jl}+k)$, $\log(\mu_{jl}+k)$. Noticing that the term $\log(\pi_{jl}+k)$ occurs in the log-likelihood if and only if $X_{il} \geq k+1$, and the term $\log(\mu_{jl}+k)$ occurs in the log-likelihood if and only if $Y_{il} \geq k+1$, we define the following associated counts for $l = 1, \ldots, D$:

$$r_{lk} = \sum_{i=1}^{t} 1_{\{X_{il} \geq k+1\}}, \quad s_{lk} = \sum_{i=1}^{t} 1_{\{Y_{il} \geq k+1\}}$$

where the index k ranges from 0 to $\max_i m_i - 1$. Recalling that $v_{ijl} = P(Z_i = j, \phi_{jl} = 1 | X_i)$ and $\nu_{ijl} = P(Z_i = j, \phi_{jl} = 0 | X_i)$, thus, Eq.(7.31) can be re-written as:

$$\begin{aligned}\mathcal{L}(\Theta) = \sum_i v_{ijl} \Bigg[& - \sum_l \sum_k s_{lk} \log(1 + k\theta_{jl}) + \sum_l \sum_k r_{lk} \log(\pi_{jl} + k\theta_{jl}) \\ & + \sum_l \sum_k s_{lk} \log\Big((1-\pi_{jl}) + k\theta_{jl}\Big) \Bigg] \\ & + \sum_i (\sum_j \nu_{ijl}) \Bigg[- \sum_l \sum_k s_{lk} \log(1 + k\lambda_l) + \sum_l \sum_k r_{lk} \log(\mu_l + k\lambda_l) \\ & + \sum_l \sum_k s_{lk} \log\Big((1-\mu_l) + k\lambda_l\Big) \Bigg] \end{aligned} \tag{7.32}$$

Then, we apply the basic minorization functions found by Zhou and Lange (see equations 2.3 and 2.4 in [232]) to the previous equation which yields the surrogate function as:

$$
\begin{aligned}
\mathcal{G}(\Theta) = \sum_i \upsilon_{ijl} \Bigg[& - \sum_l \sum_k s_{lk} \frac{k}{1 + k\theta_{jl}^n} \theta_{jl} \\
& + \sum_l \sum_k r_{lk} \left\{ \frac{\pi_{jl}^n}{\pi_{jl}^n + k\theta_{jl}^n} \log \pi_{jl} + \frac{k\theta_{jl}^n}{\pi_{jl}^n + k\theta_{jl}^n} \log \theta_{jl} \right\} \\
& + \sum_l \sum_k s_{lk} \left\{ \frac{1 - \pi_{jl}^n}{(1 - \pi_{jl}^n) + k\theta_{jl}^n} \log(1 - \pi_{jl}) + \frac{k\theta_{jl}^n}{(1 - \pi_{jl}^n) + k\theta_{jl}^n} \log k\theta_{jl}^n \right\} \Bigg] \\
& + \sum_i (\sum_j \nu_{ijl}) \Bigg[- \sum_l \sum_k s_{lk} \frac{k}{1 + k\lambda_l^n} \lambda_l \\
& + \sum_l \sum_k r_{lk} \left\{ \frac{\mu_l^n}{\mu_l^n + k\lambda_l^n} \log \mu_l + \frac{k\lambda_l^n}{\mu_l^n + k\lambda_l^n} \log \lambda_l \right\} \\
& + \sum_l \sum_k s_{lk} \left\{ \frac{1 - \mu_l^n}{(1 - \mu_l^n) + k\lambda_l^n} \log(1 - \mu_l) + \frac{k\lambda_l^n}{(1 - \mu_l^n) + k\lambda_l^n} \log k\lambda_l^n \right\} \Bigg]
\end{aligned}
\tag{7.33}
$$

Chapter 8

Conclusion

In this thesis, we have developed several approaches for high-dimensional and sparse count data clustering. Our approaches consider various mixture models of based on hierarchical Bayesian frameworks, such as the Dirichlet Compound Multinomial (DCM), Multinomial Scaled Dirichlet (MSD), Multinomial Beta-Liouville (MBL), and Generalized Dirichlet Multinomial (GDM). The proposed work is motivated by the efficiency of hierarchical Bayesian frameworks in modeling both the burstiness and overdispersion phenomena which we observe in many practical situations where the generated data are in the form of vectors of frequencies. Nevertheless, these models do not belong to the exponential family, and they are not efficient in high-dimensional spaces where many parameters need to be estimated.

In Chapter 2, we have proposed an MML-based approach to select the model that best represents the data based on a finite mixture of the exponential approximation to DCM (EDCM). The obtained results, when applied on real data, show its merits as an unsupervised learning model for clustering count data. Through a set of experiments, we have shown that the mixture of EDCM distributions with the proposed MML approach offers strong modeling capabilities for applications that involve high-dimensional and sparse count data.

A novel model called Multinomial Scaled Dirichlet (MSD), which is a composition of the multinomial and the scaled Dirichlet distributions, has been introduced in Chapter 3 for count data modeling. The approach proposed is motivated by the ability of the hierarchical Bayesian frameworks to model text data and can be used in many practical situations where the burstiness phenomenon

appears. Furthermore, we have introduced a new family of distributions (EMSD) based on the exponential family approximation of the proposed MSD. The deterministic annealing expectation-maximization (DAEM) algorithm and MML-based criterion have been proposed to estimate the parameters of the EMSD mixture model, and model selection, respectively. The effectiveness of both new mixtures was shown through extensive experiments on challenging clustering problems. The results revealed that MSD mostly outperforms the mixtures of Multinomial and DCM, and achieve comparable performance to the recently introduced MGD and MBL. On the other hand, EMSD successfully and correctly captures the burstiness phenomenon while being many times faster and computationally efficient compared to the corresponding MSD. Our unsupervised algorithm provides promising results in selecting the optimal number of clusters by optimizing the message length of the data efficiently. Then, Chapter 4 was devoted to developing hybrid generative discriminative approaches by combining appropriately the advantages of both the generative and discriminative models for modeling count data. In particular, we derived probabilistic kernels from our recently proposed finite mixture of Multinomial Scaled Dirichlet distributions. These approaches are motivated by the great number of applications that involve such types of data as well as the advantages of both SVMs and finite mixture models. The developed hybrid models are introduced as effective SVM kernels able to incorporate prior knowledge about the nature of data involved in the problem at hand and, therefore, permits good data discrimination. The achieved results suggest that an accurate classification of count data can be achieved by efficient learning of kernels from the available data.

Moreover, given that processing high-dimensional data requires significantly increasing time and space, we have introduced new exponential-family approximations to the Multinomial Beta-Liouville (MBL) and the Generalized Dirichlet Multinomial (GDM) in Chapter 5 and Chapter 6, respectively. The goal is to provide more flexible frameworks than the previously proposed EDCM that has shown to be efficient in high-dimensional spaces. We investigated different approaches for model learning: in Chapter 5, we proposed a robust learning algorithm for addressing the problems of EMBL parameters estimation and model selection simultaneously, wherein Chapter 6 Deterministic Annealing Expectation-Maximization (DAEM) algorithm and Minimum Message Length (MML) criterion have been used, respectively, for learning the parameters of the EGDM mixture and determining the number of optimal clusters. Experiments with different real-world applications

using standard and widely used datasets have shown that the proposed approximations are more efficient in terms of performance and computational complexity than their corresponding models, especially when the data are high-dimensional and sparse.

Furthermore, we considered another approach to handle high-dimensional data, namely, feature selection which is a traditional and effective approach. In Chapter 7 of this thesis, we have developed a novel unsupervised component-based feature selection approach, based on a mixture of GDM densities. Our experiments have shown that assigning a weight to each feature that shows its relevance to each class achieves better performance than assuming that all features are equally important. The efficiency of the proposed framework was demonstrated on textual, and visual datasets represented as high-dimensional frequency vectors.

There are several potential future directions that we are going to pursue. One of the most promising directions for unsupervised learning may lie in deep learning methods. Indeed, deep learning is one of the most popular methods researched now, which has shown to achieve significant results in feature representation and classification/ categorization [324–326]. Thus, a potential future work can be devoted to the development of a deep structured generative model based on the proposed mixture models as a powerful generalization to multiple layers as it has been done previously for the Gaussian mixture model [327, 328]. Other directions are towards improving the learning process of the proposed approaches. For example, a promising future work can be devoted to the development of an empirical Bayes approach to learn the model hyperparameters from the data itself or to consider another principle by deploying variational inference, especially within the formalism of the exponential family [329]. Another potential future work can be devoted to online learning via stochastic variational inference [330], and its application to classic but challenging problems such as novelty detection [37]. Moreover, estimating the feature importance in vectors that contain both continuous and discrete-valued variables can also be investigated. Finally, we can extend the proposed feature selection approach to handle streaming data as, in many applications, features may have time-varying degrees of relevance.

Bibliography

[1] M. A. T. Figueiredo and A. K. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 3, pp. 381–396, 2002.

[2] B. S. Everitt, *An introduction to finite mixture distributions*. Sage Publications Sage CA: Thousand Oaks, CA, 1996.

[3] G. McLachlan and D. Peel, *Finite mixture models*. John Wiley & Sons, 2004.

[4] M.-H. Yang and N. Ahuja, "Gaussian mixture model for human skin color and its applications in image and video databases," in *Storage and Retrieval for Image and Video Databases VII*, vol. 3656. International Society for Optics and Photonics, 1998, pp. 458–467.

[5] M. S. Handcock, A. E. Raftery, and J. M. Tantrum, "Model-based clustering for social networks," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 170, no. 2, pp. 301–354, 2007.

[6] S. Boutemedjet and D. Ziou, "Predictive approach for user long-term needs in content-based image suggestion," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 23, no. 8, pp. 1242–1253, 2012.

[7] S. M. Katz, "Distribution of content words and phrases in text and language modelling," *Natural Language Engineering*, vol. 2, no. 1, pp. 15–59, 1996.

[8] P. Puig and J. Valero, "Count data distributions: some characterizations with applications," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 332–340, 2006.

[9] R. E. Madsen, D. Kauchak, and C. Elkan, "Modeling word burstiness using the Dirichlet distribution," in *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 2005, pp. 545–552.

[10] N. Bouguila, "Clustering of count data using generalized Dirichlet multinomial distributions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 4, pp. 462–474, 2008.

[11] ——, "Count data modeling and classification using finite mixtures of distributions," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 186–198, 2011.

[12] C. Elkan, "Clustering documents with an exponential-family approximation of the Dirichlet compound multinomial distribution," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 289–296.

[13] M. J. Wainwright, M. Jordan *et al.*, "Graphical models, exponential families, and variational inference," *Foundations and Trends in Machine Learning*, vol. 1, no. 1–2, pp. 1–305, 2008.

[14] A. DasGupta, "The exponential family and statistical applications," in *Probability for Statistics and Machine Learning*. Springer, 2011, pp. 583–612.

[15] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, no. Mar, pp. 1157–1182, 2003.

[16] D. Cai, C. Zhang, and X. He, "Unsupervised feature selection for multi-cluster data," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2010, pp. 333–342.

[17] G. Chandrashekar and F. Sahin, "A survey on feature selection methods," *Computers & Electrical Engineering*, vol. 40, no. 1, pp. 16–28, 2014.

[18] R. Kohavi and D. Sommerfield, "Feature subset selection using the wrapper method: Overfitting and dynamic search space topology," in *KDD*, 1995, pp. 192–197.

[19] L. Wolf and A. Shashua, "Feature selection for unsupervised and supervised inference: The emergence of sparsity in a weight-based approach," *Journal of Machine Learning Research*, vol. 6, no. Nov, pp. 1855–1887, 2005.

[20] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge & Data Engineering*, no. 4, pp. 491–502, 2005.

[21] N. Zamzami and N. Bouguila, "MML-based approach for determining the number of topics in EDCM mixture models," in *Advances in Artificial Intelligence - 31st Canadian Conference on Artificial Intelligence, Canadian AI 2018, Toronto, ON, Canada*. Springer, 2018, pp. 211–217.

[22] ——, "Model selection and application to high-dimensional count data clustering," *Applied Intelligence*, vol. 49, no. 4, pp. 1467–1488, 2019.

[23] ——, "Text modeling using multinomial scaled dirichlet distributions," in *Mouhoub M., Sadaoui S., Ait Mohamed O., Ali M. (eds) Recent Trends and Future Technology in Applied Intelligence. IEA/AIE 2018. Lecture Notes in Computer Science, vol 10868.* Springer, 2018, pp. 69–80.

[24] ——, "A novel scaled Dirichlet-based statistical framework for count data modeling: Unsupervised learning and exponential approximation," *Pattern Recognition*, vol. 95, pp. 36–47.

[25] ——, "Hybrid generative discriminative approaches based on Multinomial scaled Dirichlet mixture models," *Applied Intelligence*, vol. 49, no. 11, pp. 3783–3800, 2019.

[26] ——, "High-dimensional count data clustering based on an exponential approximation to the Multinomial Beta-Liouville distribution," *Information Sciences*, 2019, manuscript submitted for publication.

[27] ——, "Sparse count data clustering using an exponential approximation to generalized Dirichlet Multinomial distributions," *IEEE Transactions on Neural Networks and Learning Systems*, 2019, manuscript submitted for publication.

[28] ——, "A novel MM framework for simultaneous feature selection and clustering of high-dimensional count data," *IEEE Transactions on Cybernetics*, 2019, manuscript submitted for publication.

[29] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[30] A. K. Jain and P. J. Flynn, *Image segmentation using clustering*. IEEE Press, Piscataway, NJ, 1996.

[31] H. Frigui and R. Krishnapuram, "A robust competitive clustering algorithm with applications in computer vision," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 5, pp. 450–465, 1999.

[32] M. Iwayama and T. Tokunaga, "Cluster-based text categorization: a comparison of category search strategies," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 1995, pp. 273–280.

[33] M. Sahami and D. Koller, "Using machine learning to improve information access," Ph.D. dissertation, Stanford University, Department of Computer Science, 1998.

[34] S. K. Bhatia and J. S. Deogun, "Conceptual clustering in information retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 28, no. 3, pp. 427–436, 1998.

[35] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern classification*. John Wiley & Sons, 2012.

[36] N. Bouguila and D. Ziou, "Online clustering via finite mixtures of dirichlet and minimum message length," *Engineering Applications of Artificial Intelligence*, vol. 19, no. 4, pp. 371–379, 2006.

[37] J. Zhang, Z. Ghahramani, and Y. Yang, "A probabilistic model for online document clustering with application to novelty detection," in *Advances in Neural Information Processing Systems*, 2005, pp. 1617–1624.

[38] O. Amayri and N. Bouguila, "Online news topic detection and tracking via localized feature selection," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2013, pp. 1–8.

[39] S. Singh and M. Markou, "An approach to novelty detection applied to the classification of image regions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 4, pp. 396–407, 2004.

[40] R. A. Baxter and J. J. Oliver, "Finding overlapping components with MML," *Statistics and Computing*, vol. 10, no. 1, pp. 5–16, 2000.

[41] C. S. Wallace and D. L. Dowe, "MML clustering of multi-state, Poisson, von mises circular and Gaussian distributions," *Statistics and Computing*, vol. 10, no. 1, pp. 73–83, 2000.

[42] N. Timande, M. Chandak, and M. Kamble, "Document clustering with feature selection using dirichlet process mixture model and dirichlet multinomial allocation model," *International Journal of Engineering Research and Applications*, pp. 10–16, 2014.

[43] M. Sandler, "Hierarchical mixture models: a probabilistic analysis," in *Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining (SIGKDD)*. ACM, 2007, pp. 580–589.

[44] K. A. Heller and Z. Ghahramani, "Bayesian hierarchical clustering," in *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 2005, pp. 297–304.

[45] Y. Zhao, G. Karypis, and U. Fayyad, "Hierarchical clustering algorithms for document datasets," *Data Mining and Knowledge Discovery*, vol. 10, no. 2, pp. 141–168, 2005.

[46] Ł. P. Olech and M. Paradowski, "Hierarchical gaussian mixture model with objects attached to terminal and non-terminal dendrogram nodes," in *Proceedings of the 9th International Conference on Computer Recognition Systems (CORES)*. Springer, 2016, pp. 191–201.

[47] C. M. Bishop and M. E. Tipping, "A hierarchical latent variable model for data visualization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 281–293, 1998.

[48] R. K. Blashfield, "Mixture model tests of cluster analysis: Accuracy of four agglomerative hierarchical methods." *Psychological Bulletin*, vol. 83, no. 3, p. 377, 1976.

[49] J. R. Hershey and P. A. Olsen, "Approximating the kullback leibler divergence between gaussian mixture models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4. IEEE, 2007, pp. IV–317.

[50] J. H. Jensen, D. P. Ellis, M. G. Christensen, and S. H. Jensen, "Evaluation of distance measures between gaussian mixture models of mfccs." in *ISMIR*, 2007, pp. 107–108.

[51] C. Edelbrock and B. McLaughlin, "Hierarchical cluster analysis using intraclass correlations: A mixture model study," *Multivariate Behavioral Research*, vol. 15, no. 3, pp. 299–318, 1980.

[52] A. McCallum, K. Nigam *et al.*, "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*, vol. 752, no. 1. Citeseer, 1998, pp. 41–48.

[53] N. Ueda and R. Nakano, "Deterministic annealing EM algorithm," *Neural Networks*, vol. 11, no. 2, pp. 271–282, 1998.

[54] K. W. Church and W. A. Gale, "Poisson mixtures," *Natural Language Engineering*, vol. 1, no. 2, pp. 163–190, 1995.

[55] D. Margaritis and S. Thrun, "A bayesian multiresolution independence test for continuous variables," in *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann Publishers Inc., 2001, pp. 346–353.

[56] C. S. Wallace and D. M. Boulton, "An information measure for classification," *The Computer Journal*, vol. 11, no. 2, pp. 185–194, 1968.

[57] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. 19, no. 6, pp. 716–723, 1974.

[58] J. Rissanen, "Modeling by shortest data description," *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[59] N. Bouguila and D. Ziou, "Unsupervised selection of a finite Dirichlet mixture model: an MML-based approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 8, pp. 993–1009, 2006.

[60] ——, "High-dimensional unsupervised selection and estimation of a finite generalized Dirichlet mixture model based on minimum message length," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 10, 2007.

[61] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Tackling the poor assumptions of naive bayes text classifiers," in *Proceedings of the Twentieth International Conference on Machine Learning ICML*, vol. 3, 2003, pp. 616–623.

[62] J. F. C. Kingman, *Poisson processes*. Wiley Online Library, 1993.

[63] J. E. Mosimann, "On the compound multinomial distribution, the multivariate β-distribution, and correlations among proportions," *Biometrika*, vol. 49, no. 1/2, pp. 65–82, 1962.

[64] T. P. Minka, "Estimating a Dirichlet distribution," pp. 1–13, 2003.

[65] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with bregman divergences," *Journal of Machine Learning Research*, vol. 6, no. Oct, pp. 1705–1749, 2005.

[66] S. Goldwater, T. Griffiths, and M. Johnson, "Interpolating between types and tokens by estimating power-law generators," *Advances in Neural Information Processing Systems*, vol. 18, pp. 459–467, 2006.

[67] L. D. Brown, "Fundamentals of statistical exponential families: with applications in statistical decision theory." Instiute of Mathematical Statistics, 1986.

[68] G. McLachlan and T. Krishnan, *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382.

[69] C. S. Wallace and P. R. Freeman, "Estimation and inference by compact coding," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 240–265, 1987.

[70] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[71] C. S. Wallace, *Statistical and inductive inference by minimum message length*. Springer Science & Business Media, 2005.

[72] J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices and groups*. Springer Science & Business Media, 2013, vol. 290.

[73] C. S. Wallace, "Classification by minimum-message-length inference," in *Proceedings of the International Conference on Computing and Information*. Springer, 1990, pp. 72–81.

[74] W. H. Jefferys and J. O. Berger, "Ockham's razor and Bayesian analysis," *American Scientist*, vol. 80, no. 1, pp. 64–72, 1992.

[75] N. Bouguila and D. Ziou, "Unsupervised learning of a finite discrete mixture: Applications to texture modeling and image databases summarization," *Journal of Visual Communication and Image Representation*, vol. 18, no. 4, pp. 295–309, 2007.

[76] C. C. Aggarwal and C. Zhai, "An introduction to text mining," *Mining Text Data*, pp. 1–10, 2012.

[77] Y. Lin, J. Jiang, and S. Lee, "A similarity measure for text classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 7, pp. 1575–1590, 2014.

[78] A. K. McCallum, "Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering," *http://www.cs.cmu.edu/˜mccallum/bow*, 1996.

[79] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[80] J.-F. Yao, "On recursive estimation in incomplete data models," *Statistics: A Journal of Theoretical and Applied Statistics*, vol. 34, no. 1, pp. 27–51, 2000.

[81] I. S. Dhillon and D. S. Modha, "Concept decompositions for large sparse text data using clustering," *Machine Learning*, vol. 42, no. 1, pp. 143–175, 2001.

[82] A. Banerjee and S. Basu, "Topic models over text streams: A study of batch and online unsupervised learning," in *Proceedings of the 2007 SIAM International Conference on Data Mining*. SIAM, 2007, pp. 431–436.

[83] V. Hatzivassiloglou, L. Gravano, and A. Maganti, "An investigation of linguistic features and clustering algorithms for topical document clustering," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2000, pp. 224–231.

[84] P. H. Sneath, R. R. Sokal *et al.*, *Numerical taxonomy. The principles and practice of numerical classification*. WH Freeman and Company, San Francisco, CA, 1973.

[85] S. Guha, R. Rastogi, and K. Shim, "Cure: an efficient clustering algorithm for large databases," in *Proceedings of the ACM Sigmod Record*, vol. 27, no. 2. ACM, 1998, pp. 73–84.

[86] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City*, vol. 1, no. 2, p. 1, 2007.

[87] S. Kullback and R. A. Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[88] S. Kullback, *Information theory and statistics*. Courier Corporation, 1997.

[89] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on Statistical Learning in Computer Vision, ECCV*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.

[90] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[91] C. Elkan, "Using the triangle inequality to accelerate k-means," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 147–153.

[92] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," University of Toronto, Tech. Rep., 2009.

[93] L. Fei-Fei and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 524–531.

[94] B. Yao and L. Fei-Fei, "Grouplet: A structured image representation for recognizing human and object interactions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2010, pp. 9–16.

[95] F. A. Graybill, *Matrices with applications in statistics*. Wadsworth Inc., 1983.

[96] H. Jégou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2009, pp. 1169–1176.

[97] S. Migliorati, G. S. Monti, and A. Ongaro, "E–M algorithm: an application to a mixture model for compositional data," in *Proceedings of the 44th Scientific Meeting of the Italian Statistical Society*, 2008.

[98] R. H. Lochner, "A generalized Dirichlet distribution in bayesian life testing," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 103–113, 1975.

[99] Y. Wang, Y. Y. Tang, L. Li, and X. Zheng, "Block sparse representation for pattern classification: Theory, extensions and applications," *Pattern Recognition*, vol. 88, pp. 198–209, 2019.

[100] T.-T. Wong, "Alternative prior assumptions for improving the performance of naïve Bayesian classifiers," *Data Mining and Knowledge Discovery*, vol. 18, no. 2, pp. 183–213, 2009.

[101] B. Sivazlian, "On a multivariate extension of the Gamma and Beta distributions," *SIAM Journal on Applied Mathematics*, vol. 41, no. 2, pp. 205–209, 1981.

[102] R. D. Gupta and D. S. P. Richards, "Multivariate liouville distributions," *Journal of Multivariate Analysis*, vol. 23, no. 2, pp. 233–256, 1987.

[103] G. S. Monti, G. Mateu-Figueras, and V. Pawlowsky-Glahn, *Compositional Data Analysis: Theory and Applications*. John Wiley & Sons, Chichester, UK, 2011, ch. Notes on the scaled Dirichlet distribution.

[104] J. Aitchison, "The statistical analysis of compositional data," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 139–177, 1982.

[105] E. A. Cabanlit Jr, R. N. Padua, and K. Alam, "Generalization of the direchlet distribution," *Research and Development Center Minando State University*, 2004.

[106] R. K. Hankin *et al.*, "A generalization of the Dirichlet distribution," *Journal of Statistical Software*, vol. 33, no. 11, pp. 1–18, 2010.

[107] B. S. Oboh and N. Bouguila, "Unsupervised learning of finite mixtures using scaled dirichlet distribution and its application to software modules categorization," in *2017 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2017, pp. 1085–1090.

[108] T. Krishnan and G. McLachlan, "The EM algorithm and extensions," *Wiley*, vol. 1, no. 997, pp. 58–60, 1997.

[109] S. Goldwater, M. Johnson, and T. L. Griffiths, "Interpolating between types and tokens by estimating power-law generators," in *Advances in Neural Information Processing Systems*, 2006, pp. 459–466.

[110] A. Banerjee, I. S. Dhillon, J. Ghosh, and S. Sra, "Clustering on the unit hypersphere using von Mises-Fisher distributions," *Journal of Machine Learning Research*, vol. 6, no. Sep, pp. 1345–1382, 2005.

[111] P. D. Powell, "Calculating determinants of block matrices," *arXiv preprint arXiv:1112.4379*, 2011.

[112] R. R. Larson, "Introduction to information retrieval," *Journal of the American Society for Information Science and Technology*, vol. 61, no. 4, pp. 852–853, 2010.

[113] R. Johnson and T. Zhang, "Semi-supervised convolutional neural networks for text categorization via region embedding," in *Proceedings of Advances in neural information processing systems (NIPS)*, 2015, pp. 919–927.

[114] B. Tang, S. Kay, and H. He, "Toward optimal feature selection in naive Bayes for text categorization," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 9, pp. 2508–2521, 2016.

[115] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts, "Learning word vectors for sentiment analysis," in *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*. Association for Computational Linguistics, 2011, pp. 142–150.

[116] X. Liu, B. V. Kumar, P. Jia, and J. You, "Hard negative generation for identity-disentangled facial expression recognition," *Pattern Recognition*, vol. 88, pp. 1–12, 2019.

[117] M. Valstar and M. Pantic, "Induced disgust, happiness and surprise: an addition to the MMI facial expression database," in *Proceedings of 3rd International Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect*, 2010, p. 65.

[118] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, 2010, pp. 94–101.

[119] Y. Ruichek *et al.*, "Local concave-and-convex micro-structure patterns for texture classification," *Pattern Recognition*, vol. 76, pp. 303–322, 2018.

[120] B. Julesz, "Visual pattern discrimination," *IRE transactions on Information Theory*, vol. 8, no. 2, pp. 84–92, 1962.

[121] B. Caputo, E. Hayman, and P. Mallikarjuna, "Class-specific material categorisation," in *Proceedings of Tenth IEEE International Conference on Computer Vision (ICCV)*, vol. 2. IEEE, 2005, pp. 1597–1604.

[122] L. van der Maaten and E. Postma, "Texton-based texture classification," in *Proceedings of the Belgium-Netherlands Artifical Intelligence Conference*, 2007.

[123] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh, "On the significance of real-world conditions for material classification," in *European conference on computer vision*. Springer, 2004, pp. 253–266.

[124] S. Lazebnik, C. Schmid, and J. Ponce, "A sparse texture representation using local affine regions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1265–1278, 2005.

[125] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2014, pp. 3606–3613.

[126] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: A comprehensive study," *International journal of computer vision*, vol. 73, no. 2, pp. 213–238, 2007.

[127] C. Bishop, C. M. Bishop *et al.*, *Neural networks for pattern recognition*. Oxford university press, 1995.

[128] Y. D. Rubinstein, T. Hastie *et al.*, "Discriminative vs informative learning." in *KDD*, vol. 5, 1997, pp. 49–53.

[129] V. Vapnik, *The nature of statistical learning theory*. Springer, 2013.

[130] A. Y. Ng and M. I. Jordan, "On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes," in *Advances in neural information processing systems*, 2002, pp. 841–848.

[131] R. Raina, Y. Shen, A. Mccallum, and A. Y. Ng, "Classification with hybrid generative/discriminative models," in *Advances in neural information processing systems*, 2004, pp. 545–552.

[132] B. Scholkopf and A. J. Smola, *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

[133] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Advances in neural information processing systems*, 2002, pp. 681–687.

[134] N. Christianini and J. Shawe-Taylor, "Support vector machines," *Cambridge, UK: Cambridge University Press*, vol. 93, no. 443, pp. 935–948, 2000.

[135] J. M. Moguerza, A. Muñoz *et al.*, "Support vector machines with applications," *Statistical Science*, vol. 21, no. 3, pp. 322–336, 2006.

[136] Y. Ma and G. Guo, *Support vector machines applications*. Springer, 2014.

[137] C. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer, 2006.

[138] A. Shmilovici, "Support vector machines," in *Data Mining and Knowledge Discovery Handbook*. Springer, 2010, pp. 231–247.

[139] S. S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with gaussian kernel," *Neural computation*, vol. 15, no. 7, pp. 1667–1689, 2003.

[140] H.-T. Lin and C.-J. Lin, "A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods," *submitted to Neural Computation*, vol. 3, pp. 1–32, 2003.

[141] T. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Advances in neural information processing systems*, 1999, pp. 487–493.

[142] J. Deng, X. Xu, Z. Zhang, S. Frühholz, D. Grandjean, and B. Schuller, "Fisher kernels on phase-based features for speech emotion recognition," in *Dialogues with social robots*. Springer, 2017, pp. 195–203.

[143] N. Bouguila, "Hybrid generative/discriminative approaches for proportional data modeling and classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 12, pp. 2184–2202, 2012.

[144] ——, "Deriving kernels from generalized Dirichlet mixture models and applications," *Information Processing & Management*, vol. 49, no. 1, pp. 123–137, 2013.

[145] P. Wang, L. Sun, S. Yang, and A. F. Smeaton, "Improving the classification of quantified self activities and behaviour using a fisher kernel," in *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers*. ACM, 2015, pp. 979–984.

[146] L. Van Der Maaten, "Learning discriminative fisher kernels." in *ICML*, vol. 11, 2011, pp. 217–224.

[147] N. Bouguila and O. Amayri, "A discrete mixture-based kernel for SVMs: Application to spam and image categorization," *Information Processing & Management*, vol. 45, no. 6, pp. 631–642, 2009.

[148] P. J. Moreno, P. P. Ho, and N. Vasconcelos, "A Kullback-Leibler divergence based kernel for SVM classification in multimedia applications," in *Advances in neural information processing systems*, 2004, pp. 1385–1392.

[149] A. B. Chan, N. Vasconcelos, and P. J. Moreno, "A family of probabilistic kernels based on information divergence," *Univ. California, San Diego, CA, Tech. Rep. SVCL-TR-2004-1*, 2004.

[150] N. Vasconcelos, P. Ho, and P. Moreno, "The Kullback-Leibler kernel as a framework for discriminant and localized representations for visual recognition," in *European Conference on Computer Vision*. Springer, 2004, pp. 430–441.

[151] W. D. Penny, "Kullback-Liebler divergences of normal, Gamma, Dirichlet and Wishart densities," *Wellcome Department of Cognitive Neurology*, 2001.

[152] O. Amayri and N. Bouguila, "Beyond hybrid generative discriminative learning: spherical data classification," *Pattern Analysis and Applications*, vol. 18, no. 1, pp. 113–133, 2015.

[153] A. Rényi *et al.*, "On measures of entropy and information," in *Proceedings of the Fourth*

Berkeley Symposium on Mathematical Statistics and Probability. The Regents of the University of California, 1961.

[154] J. Lin, "Divergence measures based on the shannon entropy," *IEEE Transactions on Information theory*, vol. 37, no. 1, pp. 145–151, 1991.

[155] W. M. Campbell, D. E. Sturim, and D. A. Reynolds, "Support vector machines using GMM super vectors for speaker verification," *IEEE Signal Process. Lett.*, vol. 13, no. 5, pp. 308–311, 2006.

[156] F. Pérez-Cruz, "Kullback-Leibler divergence estimation of continuous distributions," in *Information Theory, 2008. ISIT 2008. IEEE International Symposium on*. IEEE, 2008, pp. 1666–1670.

[157] N. Bouguila, "Bayesian hybrid generative discriminative learning based on finite liouville mixture models," *Pattern Recognition*, vol. 44, no. 6, pp. 1183–1200, 2011.

[158] T. Bdiri and N. Bouguila, "Bayesian learning of inverted Dirichlet mixtures for SVM kernels generation," *Neural Computing and Applications*, vol. 23, no. 5, pp. 1443–1458, 2013.

[159] B. S. Oboh and N. Bouguila, "Unsupervised learning of finite mixtures using scaled Dirichlet distribution and its application to software modules categorization," in *Proccedings of the 2017 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, 2017, pp. 1085–1090.

[160] A. P. Dempster, "Maximum likelihood estimation from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 39, pp. 1–38, 1977.

[161] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery*, vol. 2, no. 2, pp. 121–167, 1998.

[162] R. A. Berk, "Support vector machines," in *Statistical Learning from a Regression Perspective*. Springer, 2016, pp. 291–310.

[163] A. Agarwal, H. DaumÃ *et al.*, "Generative kernels for exponential families," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 85–92.

[164] T. Jebara, "Images as bags of pixels." in *ICCV*, 2003, pp. 265–272.

[165] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Tenth IEEE International Conference on Computer Vision (ICCV)*, vol. 2. IEEE, 2005, pp. 1458–1465.

[166] T. Jebara, R. Kondor, and A. Howard, "Probability product kernels," *Journal of Machine Learning Research*, vol. 5, no. Jul, pp. 819–844, 2004.

[167] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories," *Computer vision and Image understanding*, vol. 106, no. 1, pp. 59–70, 2007.

[168] V. Ferrari, T. Tuytelaars, and L. Van Gool, "Object detection by contour segment networks," in *European conference on computer vision*. Springer, 2006, pp. 14–28.

[169] J. Winn, A. Criminisi, and T. Minka, "Object categorization by learned universal visual dictionary," in *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, vol. 2. IEEE, 2005, pp. 1800–1807.

[170] H. Mureşan and M. Oltean, "Fruit recognition from images using deep learning," *Acta Universitatis Sapientiae, Informatica*, vol. 10, no. 1, pp. 26–42, 2018.

[171] S.-K. Chang and A. Hsu, "Image information systems: where do we go from here?" *IEEE transactions on Knowledge and Data Engineering*, vol. 4, no. 5, pp. 431–442, 1992.

[172] A. Bosch, X. Muñoz, and R. Martí, "Which is the best way to organize/classify images by content?" *Image and vision computing*, vol. 25, no. 6, pp. 778–791, 2007.

[173] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014, pp. 487–495.

[174] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *IEEE transactions on communication technology*, vol. 15, no. 1, pp. 52–60, 1967.

[175] N. Zamzami and N. Bouguila, "Consumption behavior prediction using hierarchical Bayesian frameworks," in *2018 First International Conference on Artificial Intelligence for Industries (AI4I)*. IEEE, 2018, pp. 31–34.

[176] G. J. McLachlan and K. E. Basford, *Mixture models: Inference and applications to clustering*. Marcel Dekker, 1988, vol. 84.

[177] C. Hennig, "Methods for merging Gaussian mixture components," *Advances in data analysis and classification*, vol. 4, no. 1, pp. 3–34, 2010.

[178] F. Carcillo, Y.-A. Le Borgne, O. Caelen, Y. Kessaci, F. Oblé, and G. Bontempi, "Combining unsupervised and supervised learning in credit card fraud detection," *Information Sciences*, 2019.

[179] D. M. Titterington, A. F. Smith, and U. E. Makov, *Statistical analysis of finite mixture distributions*. Wiley,, 1985.

[180] I. D. Gebru, X. Alameda-Pineda, F. Forbes, and R. Horaud, "EM algorithms for weighted-data clustering with application to audio-visual scene analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 12, pp. 2402–2415, 2016.

[181] T. Hastie and R. Tibshirani, "Discriminant analysis by Gaussian mixtures," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 155–176, 1996.

[182] S. J. Roberts, D. Husmeier, I. Rezek, and W. Penny, "Bayesian approaches to Gaussian mixture modeling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 11, pp. 1133–1142, 1998.

[183] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "SMEM algorithm for mixture models," in *Advances in neural information processing systems*, 1999, pp. 599–605.

[184] P. Meinicke and H. Ritter, "Resolution-based complexity control for Gaussian mixture models," *Neural computation*, vol. 13, no. 2, pp. 453–475, 2001.

[185] K. Rose, "Deterministic annealing for clustering, compression, classification, regression, and related optimization problems," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2210–2239, 1998.

[186] M.-S. Yang, C.-Y. Lai, and C.-Y. Lin, "A robust EM clustering algorithm for Gaussian mixture models," *Pattern Recognition*, vol. 45, no. 11, pp. 3950–3961, 2012.

[187] G. Celeux, S. Chrétien, F. Forbes, and A. Mkhadri, "A component-wise EM algorithm for mixtures," *Journal of Computational and Graphical Statistics*, vol. 10, no. 4, pp. 697–712, 2001.

[188] C. E. Shannon, W. Weaver, and A. W. Burks, "The mathematical theory of communication," 1951.

[189] J. M. Bernardo and A. F. Smith, "Bayesian theory," 2001.

[190] S. Amari and H. Nagaoka, *Methods of information geometry*. American Mathematical Soc., 2007, vol. 191.

[191] C. Robert, "Generalized linear models and the exponential family," in *Machine learning, a probabilistic perspective*. Taylor & Francis, 2014, pp. 281–305.

[192] K. Poortema, "On modelling overdispersion of counts," *Statistica Neerlandica*, vol. 53, no. 1, pp. 5–20, 1999.

[193] B. Liu, "Sentiment analysis and opinion mining," *Synthesis lectures on human language technologies*, vol. 5, no. 1, pp. 1–167, 2012.

[194] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.

[195] X. Zhang, J. Zhao, and Y. LeCun, "Character-level convolutional networks for text classification," in *Advances in neural information processing systems*, 2015, pp. 649–657.

[196] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.

[197] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *arXiv preprint arXiv:1607.01759*, 2016.

[198] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for natural language processing," *arXiv preprint arXiv:1606.01781*, vol. 2, 2016.

[199] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[200] E. Sokic and S. Konjicija, "Phase preserving fourier descriptor for shape-based image retrieval," *Signal Processing: Image Communication*, vol. 40, pp. 82–96, 2016.

[201] X. Shu and X.-J. Wu, "A novel contour descriptor for 2d shape matching and its application to image retrieval," *Image and vision Computing*, vol. 29, no. 4, pp. 286–294, 2011.

[202] A. Frome, Y. Singer, F. Sha, and J. Malik, "Learning globally-consistent local distance functions for shape-based image retrieval and classification," in *2007 IEEE 11th International Conference on Computer Vision*. IEEE, 2007, pp. 1–8.

[203] L. J. Latecki, R. Lakämper and U. Eckhardt, "Shape Descriptors for Non-rigid Shapes with a Single Closed Contour," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000, pp. 424–429.

[204] S. Belongie, J. Malik and J. Puzicha, "Shape Matching and Objects Recognition Using Shape Contexts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 4, pp. 509–522, 2002.

[205] O. Tursun and S. Kalkan, "METU dataset: A big dataset for benchmarking trademark retrieval," in *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*. IEEE, 2015, pp. 514–517.

[206] L. J. Latecki, R. Lakamper, and T. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, vol. 1. IEEE, 2000, pp. 424–429.

[207] J. OSIKAR, "Computer vision classification of leaves from swedish trees," *Linkoping: Linkoping University*, 2001.

[208] S. Zhang, Y.-K. Lei, and Y.-H. Wu, "Semi-supervised locally discriminant projection for classification and recognition," *Knowledge-Based Systems*, vol. 24, no. 2, pp. 341–346, 2011.

[209] R. Hu, W. Jia, H. Ling, and D. Huang, "Multiscale distance matrix for fast plant leaf recognition," *IEEE Transactions on Image Processing*, vol. 21, no. 11, pp. 4667–4672, 2012.

[210] S. Zhang, Y. Lei, C. Zhang, and Y. Hu, "Semi-supervised orthogonal discriminant projection for plant leaf classification," *Pattern Analysis and Applications*, vol. 19, no. 4, pp. 953–961, 2016.

[211] C. Zhao, S. S. Chan, W.-K. Cham, and L. Chu, "Plant identification using leaf shapes—a pattern counting approach," *Pattern Recognition*, vol. 48, no. 10, pp. 3203–3215, 2015.

[212] B. Wang, D. Brown, Y. Gao, and J. La Salle, "March: Multiscale-arch-height description for mobile retrieval of leaf images," *Information Sciences*, vol. 302, pp. 132–148, 2015.

[213] I. Laptev, "On space-time interest points," *International journal of computer vision*, vol. 64, no. 2-3, pp. 107–123, 2005.

[214] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th ACM international conference on Multimedia*. ACM, 2007, pp. 357–360.

[215] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2008, pp. 1–8.

[216] A. Patron-Perez, M. Marszalek, A. Zisserman, and I. D. Reid, "High five: Recognising human interactions in tv shows." in *BMVC*, vol. 1. Citeseer, 2010, p. 2.

[217] A. Patron-Perez, M. Marszalek, I. Reid, and A. Zisserman, "Structured learning of human interactions in tv shows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2441–2453, 2012.

[218] Y. Yang and M. Shah, "Complex events detection using data-driven concepts," in *European Conference on Computer Vision*. Springer, 2012, pp. 722–735.

[219] A. Gaidon, Z. Harchaoui, and C. Schmid, "Activity representation with motion hierarchies," *International journal of computer vision*, vol. 107, no. 3, pp. 219–238, 2014.

[220] H. Wang, D. Oneata, J. Verbeek, and C. Schmid, "A robust and efficient video representation for action recognition," *International Journal of Computer Vision*, vol. 119, no. 3, pp. 219–238, 2016.

[221] S. Ma, J. Zhang, S. Sclaroff, N. Ikizler-Cinbis, and L. Sigal, "Space-time tree ensemble for action recognition and localization," *International Journal of Computer Vision*, vol. 126, no. 2-4, pp. 314–332, 2018.

[222] S. Yin and J. Yin, "Tuning kernel parameters for SVM based on expected square distance ratio," *Information Sciences*, vol. 370, pp. 92–102, 2016.

[223] A. M. Martinez, "The AR face database," *CVC Technical Report24*, 1998.

[224] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of the Second IEEE Workshop on Applications of Computer Vision*. IEEE, 1994, pp. 138–142.

[225] T. Jebara and R. Kondor, "Bhattacharyya and expected likelihood kernels," in *Learning theory and kernel machines*. Springer, 2003, pp. 57–71.

[226] L. Zhang, M. Yang, and X. Feng, "Sparse representation or collaborative representation: Which helps face recognition?" in *2011 International conference on computer vision*. IEEE, 2011, pp. 471–478.

[227] M. A. Borgi, M. El'Arbi, D. Labate, and C. B. Amar, "Face, gender and race classification using multi-regularized features learning," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 5277–5281.

[228] F. Juefei-Xu, E. Verma, P. Goel, A. Cherodian, and M. Savvides, "Deepgender: Occlusion and low resolution robust facial gender classification via progressively trained convolutional neural networks with attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2016, pp. 68–77.

[229] T. Li, S. Ma, and M. Ogihara, "Document clustering via adaptive subspace iteration," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 2004, pp. 218–225.

[230] J. Johnston and G. Hamerly, "Improving simpoint accuracy for small simulation budgets with EDCM clustering," *Worksh. on Statistical and Machine learning approaches to ARchitectures and compilaTion (SMART08)*, 2008.

[231] R. Cummins, J. H. Paik, and Y. Lv, "A pólya urn document language model for improved information retrieval," *ACM Transactions on Information Systems (TOIS)*, vol. 33, no. 4, p. 21, 2015.

[232] H. Zhou and K. Lange, "MM algorithms for some discrete multivariate distributions," *Journal of Computational and Graphical Statistics*, vol. 19, no. 3, pp. 645–665, 2010.

[233] Y. Zhang, H. Zhou, J. Zhou, and W. Sun, "Regression models for multivariate count data," *Journal of Computational and Graphical Statistics*, vol. 26, no. 1, pp. 1–13, 2017.

[234] G. Gan, C. Ma, and J. Wu, *Data clustering: theory, algorithms, and applications*. Siam, 2007, vol. 20.

[235] M. Al Mashrgy, T. Bdiri, and N. Bouguila, "Robust simultaneous positive data clustering and unsupervised feature selection using generalized inverted Dirichlet mixture models," *Knowledge-Based Systems*, vol. 59, pp. 182–195, 2014.

[236] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, vol. 97, no. 458, pp. 611–631, 2002.

[237] J. W. Comley and D. L. Dowe, "Minimum message length, MDL and generalised bayesian networks with asymmetric languages," *Advances in Minimum Description Length: Theory and Applications (MDL Handbook)*, 2005.

[238] C. S. Wallace, "Intrinsic classification of spatially correlated data," *The Computer Journal*, vol. 41, no. 8, pp. 602–611, 1998.

[239] C. Silvestre, M. G. Cardoso, and M. A. Figueiredo, "Identifying the number of clusters in discrete mixture models," *arXiv preprint arXiv:1409.7419*, 2014.

[240] R. J. Connor and J. E. Mosimann, "Concepts of independence for proportions with a generalization of the Dirichlet distribution," *Journal of the American Statistical Association*, vol. 64, no. 325, pp. 194–206, 1969.

[241] K. L. Caballero, J. Barajas, and R. Akella, "The generalized Dirichlet distribution in enhanced topic detection," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*. ACM, 2012, pp. 773–782.

[242] T.-T. Wong, "A Bayesian approach employing generalized Dirichlet priors in predicting microchip yields," *Journal of the Chinese Institute of Industrial Engineers*, vol. 22, no. 3, pp. 210–217, 2005.

[243] ——, "Generalized Dirichlet priors for naïve Bayesian classifiers with multinomial models in document classification," *Data Mining and Knowledge Discovery*, vol. 28, no. 1, pp. 123–144, 2014.

[244] T. Wong, "Generalized Dirichlet distribution in Bayesian analysis," *Applied Mathematics and Computation*, vol. 97, no. 2-3, pp. 165–181, 1998.

[245] P. Lewy, "A generalized Dirichlet distribution accounting for singularities of the variables," *Biometrics*, pp. 1394–1409, 1996.

[246] P. Cerchiello and P. Giudici, "Dirichlet compound Multinomials statistical models," *Applied Mathematics*, vol. 3, no. 12, p. 2089, 2012.

[247] C. D. Manning, P. Raghavan, H. Schütze *et al.*, *Introduction to information retrieval*. Cambridge University Press, 2008, vol. 1, no. 1.

[248] M. Craven, A. McCallum, D. PiPasquo, T. Mitchell, and D. Freitag, "Learning to extract symbolic knowledge from the world wide web," Carnegie-mellon University, Tech. Rep., 1998.

[249] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2007, pp. 1–8.

[250] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *IEEE conference on Computer vision and pattern recognition (CVPR)*. IEEE, 2010, pp. 3485–3492.

[251] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *International Journal of Computer Vision*, vol. 42, no. 3, pp. 145–175, 2001.

[252] Y. Song, L. Goncalves, and P. Perona, "Unsupervised learning of human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 7, pp. 814–827, 2003.

[253] T. Elguebaly and N. Bouguila, "Improving codebook generation for action recognition using a mixture of asymmetric Gaussians," in *IEEE Symposium on Computational Intelligence for Multimedia, Signal and Vision Processing (CIMSIVP)*. IEEE, 2014, pp. 1–7.

[254] Y. Wang and G. Mori, "Human action recognition by semilatent topic models," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 10, pp. 1762–1774, 2009.

[255] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"," in *IEEE conference on Computer vision and pattern recognition (CVPR)*. IEEE, 2009, pp. 1996–2003.

[256] J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," *Journal of Machine Learning Research*, vol. 5, no. Aug, pp. 845–889, 2004.

[257] H. Liu, X. Wu, and S. Zhang, "Feature selection using hierarchical feature clustering," in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*. ACM, 2011, pp. 979–984.

[258] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Computational Biology and Chemistry*, vol. 32, no. 1, pp. 29–38, 2008.

[259] C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe, "A survey on filter techniques for feature selection in gene expression microarray analysis," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 9, no. 4, pp. 1106–1119, 2012.

[260] J. Tang and H. Liu, "Feature selection with linked data in social media," in *Proceedings of the 2012 SIAM International Conference on Data Mining*. SIAM, 2012, pp. 118–128.

[261] ——, "An unsupervised feature selection framework for social media data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 12, pp. 2914–2927, 2014.

[262] L. Liu, L. Shao, and P. Rockett, "Boosted key-frame selection and correlated pyramidal motion-feature representation for human action recognition," *Pattern Recognition*, vol. 46, no. 7, pp. 1810–1818, 2013.

[263] C.-H. Lin, H.-Y. Chen, and Y.-S. Wu, "Study of image retrieval and classification based on

adaptive features using genetic algorithm feature selection," *Expert Systems with Applications*, vol. 41, no. 15, pp. 6611–6621, 2014.

[264] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions on Neural Networks*, vol. 5, no. 4, pp. 537–550, 1994.

[265] Z. Zeng, X. Wang, J. Zhang, and Q. Wu, "Semi-supervised feature selection based on local discriminative information," *Neurocomputing*, vol. 173, pp. 102–109, 2016.

[266] X. Chen, G. Yuan, F. Nie, and J. Z. Huang, "Semi-supervised feature selection via rescaled linear regression." in *IJCAI*, vol. 2017, 2017, pp. 1525–1531.

[267] M. H. Law, M. A. Figueiredo, and A. K. Jain, "Simultaneous feature selection and clustering using mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1154–1166, 2004.

[268] N. Bouguila, "A model-based approach for discrete data clustering and feature weighting using MAP and stochastic complexity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 12, pp. 1649–1664, 2009.

[269] M. Luo, F. Nie, X. Chang, Y. Yang, A. G. Hauptmann, and Q. Zheng, "Adaptive unsupervised feature selection with structure regularization," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 4, pp. 944–956, 2017.

[270] Z. Li, J. Liu, X. Zhu, T. Liu, and H. Lu, "Image annotation using multi-correlation probabilistic matrix factorization," in *Proceedings of the 18th ACM International Conference on Multimedia*. ACM, 2010, pp. 1187–1190.

[271] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu, "Clustering-guided sparse structural learning for unsupervised feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2138–2150, 2014.

[272] X. Hong, H. Li, P. Miller, J. Zhou, L. Li, D. Crookes, Y. Lu, X. Li, and H. Zhou, "Component-based feature saliency for clustering," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[273] J. M. Ortega and W. C. Rheinboldt, *Iterative solution of nonlinear equations in several variables*. Siam, 1970, vol. 30.

[274] T. T. Wu, K. Lange *et al.*, "The MM alternative to EM," *Statistical Science*, vol. 25, no. 4, pp. 492–505, 2010.

[275] M. W. Graham and D. J. Miller, "Unsupervised learning of parsimonious mixtures on large spaces with integrated feature and component selection," *IEEE Transactions on Signal Processing*, vol. 54, no. 4, pp. 1289–1303, 2006.

[276] X. Wu, B. Jiang, K. Yu, C. Miao, and H. Chen, "Accurate markov boundary discovery for causal feature selection," *IEEE transactions on cybernetics*, 2019.

[277] C. Liu, C.-T. Zheng, S. Wu, Z. Yu, and H.-S. Wong, "Multitask feature selection by graph-clustered feature sharing," *IEEE transactions on cybernetics*, 2018.

[278] H. Wu, T. Liu, and J. Xie, "Fine-grained product feature extraction in chinese reviews," in *2017 International Conference on Computing Intelligence and Information System (CIIS)*. IEEE, 2017, pp. 327–331.

[279] I. Marquetti, J. V. Link, A. L. G. Lemes, M. B. dos Santos Scholz, P. Valderrama, and E. Bona, "Partial least square with discriminant analysis and near infrared spectroscopy for evaluation of geographic and genotypic origin of arabica coffee," *Computers and Electronics in Agriculture*, vol. 121, pp. 313–319, 2016.

[280] Z. Fan, Y. Xu, W. Zuo, J. Yang, J. Tang, Z. Lai, and D. Zhang, "Modified principal component analysis: An integration of multiple similarity subspace models," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 8, pp. 1538–1552, 2014.

[281] H. Zhao, Z. Wang, and F. Nie, "A new formulation of linear discriminant analysis for robust dimensionality reduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 4, pp. 629–640, 2018.

[282] C. Bouveyron and C. Brunet-Saumard, "Model-based clustering of high-dimensional data: A review," *Computational Statistics & Data Analysis*, vol. 71, pp. 52–78, 2014.

[283] M. Dash and H. Liu, "Feature selection for clustering," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer, 2000, pp. 110–121.

[284] Y. Wang and L. Feng, "A new hybrid feature selection based on multi-filter weights and multi-feature weights," *Applied Intelligence*, pp. 1–25, 2019.

[285] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1, no. 10.

[286] H. Liu and H. Motoda, *Computational methods of feature selection*. CRC Press, 2007.

[287] M. Dash, K. Choi, P. Scheuermann, and H. Liu, "Feature selection for clustering-a filter solution," in *Proceedings of 2002 IEEE International Conference on Data Mining*. IEEE, 2002, pp. 115–122.

[288] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

[289] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[290] M. M. Kabir, M. M. Islam, and K. Murase, "A new wrapper feature selection approach using neural network," *Neurocomputing*, vol. 73, no. 16-18, pp. 3273–3283, 2010.

[291] J. Apolloni, G. Leguizamón, and E. Alba, "Two hybrid wrapper-filter feature selection algorithms applied to high-dimensional microarray experiments," *Applied Soft Computing*, vol. 38, pp. 922–932, 2016.

[292] M. Moradkhani, A. Amiri, M. Javaherian, and H. Safari, "A hybrid algorithm for feature subset selection in high-dimensional datasets using FICA and IWSSr algorithm," *Applied Soft Computing*, vol. 35, pp. 123–135, 2015.

[293] Z. Li, J. Liu, Y. Yang, X. Zhou, and H. Lu, "Clustering-guided sparse structural learning for unsupervised feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2138–2150, 2013.

[294] F. Bouillot, P. N. Hai, N. Béchet, S. Bringay, D. Ienco, S. Matwin, P. Poncelet, M. Roche, and M. Teisseire, "How to extract relevant knowledge from tweets?" in *International Workshop on Information Search, Integration, and Personalization*. Springer, 2012, pp. 111–120.

[295] D. Mladenic and M. Grobelnik, "Feature selection for unbalanced class distribution and naive bayes," in *ICML*, vol. 99, 1999, pp. 258–267.

[296] M. F. Caropreso, S. Matwin, and F. Sebastiani, "A learner-independent evaluation of the usefulness of statistical phrases for automated text categorization," *Text Databases and Document Management: Theory and Practice*, vol. 5478, pp. 78–102, 2001.

[297] Y. Li, C. Luo, and S. M. Chung, "Text clustering with feature selection by using statistical data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 5, pp. 641–652, 2008.

[298] L. Galavotti, F. Sebastiani, and M. Simi, "Experiments on the use of feature selection and negative evidence in automated text categorization," in *International Conference on Theory and Practice of Digital Libraries*. Springer, 2000, pp. 59–68.

[299] L. Talavera, "Feature selection as a preprocessing step for hierarchical clustering," in *ICML*, vol. 99. Citeseer, 1999, pp. 389–397.

[300] X. He, D. Cai, and P. Niyogi, "Laplacian score for feature selection," in *Advances in Neural Information Processing Systems*, 2006, pp. 507–514.

[301] A. Dasgupta, P. Drineas, B. Harb, V. Josifovski, and M. W. Mahoney, "Feature selection methods for text classification," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2007, pp. 230–239.

[302] S. Adams and P. A. Beling, "A survey of feature selection methods for Gaussian mixture models and hidden Markov models," *Artificial Intelligence Review*, pp. 1–41, 2017.

[303] S. Boutemedjet, N. Bouguila, and D. Ziou, "A hybrid feature extraction selection approach for high-dimensional non-Gaussian data clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 8, pp. 1429–1443, 2008.

[304] W. Fan, N. Bouguila, and D. Ziou, "Unsupervised hybrid feature extraction selection for high-dimensional non-Gaussian data clustering with variational inference," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 7, pp. 1670–1685, 2012.

[305] S. Vaithyanathan and B. Dom, "Generalized model selection for unsupervised learning in high dimensions," in *Advances in Neural Information Processing Systems*, 2000, pp. 970–976.

[306] X. Wang and A. Kabán, "Model-based estimation of word saliency in text," in *International Conference on Discovery Science*. Springer, 2006, pp. 279–290.

[307] Y.-m. Cheung and H. Zeng, "A maximum weighted likelihood approach to simultaneous model selection and feature weighting in Gaussian mixture," in *International Conference on Artificial Neural Networks*. Springer, 2007, pp. 78–87.

[308] C.-Y. Tsai and C.-C. Chiu, "Developing a feature weight self-adjustment mechanism for a K-means clustering algorithm," *Computational Statistics & Data Analysis*, vol. 52, no. 10, pp. 4658–4672, 2008.

[309] J. B. Haldane, "The fitting of binomial distributions," *Annals of Eugenics*, vol. 11, no. 1, pp. 179–181, 1941.

[310] N. T. Bailey, "The mathematical theory of epidemics," Tech. Rep., 1957.

[311] D. Griffiths, "Maximum likelihood estimation for the beta-binomial distribution and an application to the household distribution of the total number of cases of a disease," *Biometrics*, pp. 637–648, 1973.

[312] P. Pudil, J. Novovičová, N. Choakjarernwanit, and J. Kittler, "Feature selection based on the approximation of class densities by finite mixtures of special type," *Pattern Recognition*, vol. 28, no. 9, pp. 1389–1398, 1995.

[313] H. D. Nguyen, "An introduction to Majorization-Minimization algorithms for machine learning and statistical estimation," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 7, no. 2, p. e1198, 2017.

[314] G.-L. Tian, Y. Liu, M.-L. Tang, and T. Li, "A novel MM algorithm and the mode-sharing method in bayesian computation for the analysis of general incomplete categorical data," *Computational Statistics & Data Analysis*, 2019.

[315] J. Novovičová and A. Malik, "Application of multinomial mixture model to text classification," in *Iberian Conference on Pattern Recognition and Image Analysis*. Springer, 2003, pp. 646–653.

[316] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Eleventh international aaai conference on web and social media*, 2017.

[317] E. G. Ortiz and B. C. Becker, "Face recognition for web-scale datasets," *Computer Vision and Image Understanding*, vol. 118, pp. 153–170, 2014.

[318] N. Kumar, A. Berg, P. N. Belhumeur, and S. Nayar, "Describable visual attributes for face verification and image search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 10, pp. 1962–1977, 2011.

[319] G. B. Huang, M. Mattar, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database forstudying face recognition in unconstrained environments," 2008.

[320] Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5810–5818.

[321] K. Ricanek and T. Tesafaye, "MORPH: A longitudinal image database of normal adult age-progression," in *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*. IEEE, 2006, pp. 341–345.

[322] G. Guo and C. Zhang, "A study on cross-population age estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4257–4263.

[323] Z. He, X. Li, Z. Zhang, F. Wu, X. Geng, Y. Zhang, M.-H. Yang, and Y. Zhuang, "Data-dependent label distribution learning for age estimation," *IEEE Transactions on Image processing*, vol. 26, no. 8, pp. 3846–3858, 2017.

[324] D. Erhan, Y. Bengio, A. Courville, P.-A. Manzagol, P. Vincent, and S. Bengio, "Why does unsupervised pre-training help deep learning?" *Journal of Machine Learning Research*, vol. 11, no. Feb, pp. 625–660, 2010.

[325] G. Song and Q. Dai, "A novel double deep elms ensemble system for time series forecasting," *Knowledge-Based Systems*, vol. 134, pp. 31–49, 2017.

[326] X. Han and Q. Dai, "Batch-normalized mlpconv-wise supervised pre-training network in network," *Applied Intelligence*, vol. 48, no. 1, pp. 142–155, 2018.

[327] A. Van den Oord and B. Schrauwen, "Factoring variations in natural images with deep gaussian mixture models," in *Advances in Neural Information Processing Systems*, 2014, pp. 3518–3526.

[328] E. Variani, E. McDermott, and G. Heigold, "A Gaussian mixture model layer jointly optimized with discriminative features within a deep neural network architecture," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4270–4274.

[329] H. Attias, "A variational baysian framework for graphical models," in *Advances in Neural Information Processing Systems*, 2000, pp. 209–215.

[330] M. Hoffman, D. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.